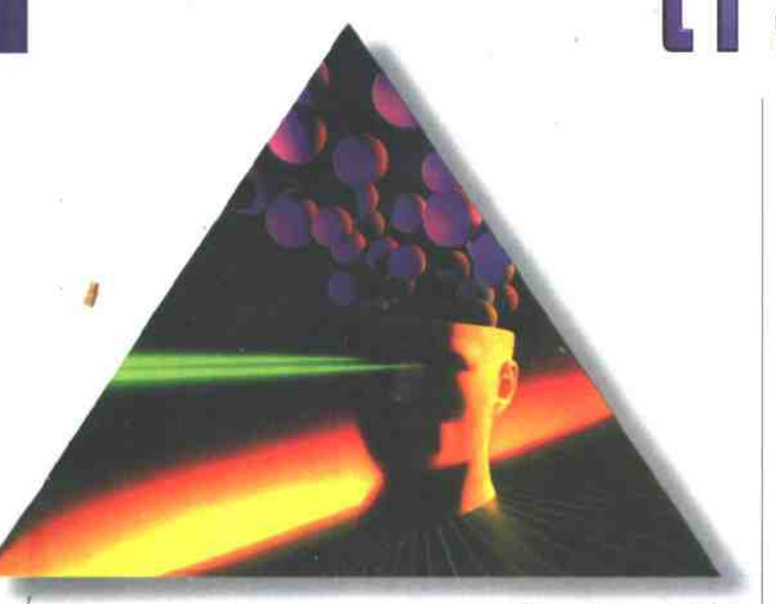
控制系统MATLAB计算及仿真

MATLAB



黄忠霖　编著

国防工业出版社
National Defence Industry Press
http://www.ndip.com.cn

控制系统 MATLAB 计算及仿真

黄忠霖　编著

图书在版编目（CIP）数据

控制系统MATLAB计算及仿真/黄忠霖编著. —北京：国防工业出版社，2001.11
ISBN 7-118-02603-4

Ⅰ. 控… Ⅱ. 黄… Ⅲ. 计算机辅助计算—软件包，MATLAB Ⅳ. TP391.75

中国版本图书馆CIP数据核字（2001）第042533号

国防工业出版社出版发行
（北京市海淀区紫竹院南路23号）
（邮政编码 100044）
北京奥隆印刷厂印刷
新华书店经售

*

开本787×1092 1/16 印张33 3/4 784千字
2001年11月第1版 2001年11月北京第1次印刷
印数：1—3000册 定价：45.00元

前　言

自从 Mathworks 软件公司开发的数值计算软件 MATLAB 问世以来，就引起了国内外学者的广泛关注。MATLAB 推出后不久，即风行美国、西欧，流传世界。特别是随着 MATLAB 控制工具箱的开发、丰富与逐步完善，使得越来越多的从事自动控制专业的科技工作者接触、了解并熟悉使用它。

自动控制领域里有大量繁琐的计算与仿真曲线绘制任务。计算机被广泛应用以后，人们有了先进的计算工具，但又不得不与计算机程序打交道，要求必须熟悉计算机语言的语句命令、语法规则、程序编制与调试的规则方法等，而要把一个功能很强的高级语言学会、学好并非易事。

MATLAB（Matrix Laboratory，即“矩阵实验室”）是当今世界上最优秀的数值计算软件。MATLAB 强大的计算功能；丰富、方便的图形功能；适用范围广；编程效率高，扩充能力强；语句简单，易学易用；功能齐备的自动控制软件工具包等优点，正是它广为流传的原因。特别是当今控制界的很多权威专家，在各自从事的控制领域里开发了具有特殊功能的软件工具箱，使得 MATLAB 从一个数值运算软件成为自动控制计算与仿真的强有力工具。MATLAB 的控制工具箱，已覆盖了控制系统的各个领域，每一个工具箱都是当今世界上该控制领域里最顶尖、最优秀的计算与仿真软件。所以 MATLAB 已经成为国际、国内控制领域内最流行的、被广泛采用的控制系统计算、仿真与计算机辅助设计软件。

MATLAB 的研究工作在国内也已得到很大发展。很多高校都开展了有关 MATLAB 内容的教学及其相关的科研工作，并取得了一批可喜的科研成果，极大地推动了控制领域 MATLAB 的仿真研究。

本书就是本着把最好的计算仿真软件 MATLAB 与自动控制两者结合起来的宗旨编写的。如果没有 MATLAB 的基础，直接介绍控制系统 MATLAB 计算及仿真，可能有很多读者对如何进入 MATLAB，如何运行程序，如何仿真还都不知晓。这样，还是得买一本有关 MATLAB 语言的书，先学学程序设计基础及语法规则。这不仅不方便，而且也不经济。基于这个考虑，本书内容包括 MATLAB 程序设计语言基础与控制系统 MATLAB 计算及仿真两大部分：第一部分介绍 MATLAB 用于控制系统计算及仿真的有关知识，包括 MATLAB 数值运算基础；M 文件与 MATLAB 函数；MATLAB 程序设计基础与 MATLAB 的文字处理工具 Notebook 等，并具体介绍了仿真集成环境 SIMULINK。这部分已独立构成 MATLAB 的完整体系。附录 A 列出了 MATLAB 程序设计中经常使用的各种函数命令，且按英语字母顺序索引，以方便读者查阅。

本书第二部分内容涵盖了经典控制理论、电力拖动控制系统、位置随动系统、过程控制系统、现代控制理论等方面，介绍控制系统仿真基础，自动控制系统的 MATLAB

计算及仿真，自动控制系统设计，状态空间分析的 MATLAB 实现，以及最优控制系统的 MATLAB 计算及仿真。附录 B 也按英语字母顺序列出了 MATLAB 系统提供的 TOOLBOX 控制工具箱函数命令。函数的内容极为丰富，都是用于自动控制方面的。

另外，在扩展 MATLAB 函数库与创新方面，作者对于典型输入信号响应稳态误差的计算与仿真、阶跃给定输入响应性能指标的计算与仿真以及控制系统校正设计等方面开发了 12 个 MATLAB 函数，供读者进行 MATLAB 计算与仿真时使用。

在叙述方法上，每个章节先对与自动控制有关的基本概念进行简要的介绍，然后介绍 MATLAB 有关函数并举例。这样既提出并引入了自动控制的基本问题，又提供解决问题的工具，并辅以算例帮助读者理解与消化。

本书示例丰富，大部分示例取材于自动控制理论、电力拖动控制、位置随动控制、生产过程控制、现代控制理论等方面的经典教科书，还有一部分取自实际工程的课题，有些课题是作者参与完成的工程项目。本书各章节的示例程序都经作者反复调试，既能够在 MATLAB 里运行，也能够在 Notebook 中运行（除开一些带鼠标操作的图形函数命令必须在 MATLAB 命令窗口里运行之外）。

本书可作为自动控制及其相关专业、机械设备及自动化专业或机电一体化专业的学生教材，也可作为自动控制领域科技人员的参考用书。

在本书编写过程中，黄京、王冠先、毛瑞刚绘制了部分插图，黄京、陈蓓蕾、吴江陵录入了部分文稿，许德阳、张慧峰、张育林协助整理了部分附录，刘福、关大海、黄少铭、马男华录入并校对了部分程序，彭达成、姜迎春、李德瑞、戴云南、田园亮、夏天浩核对了全书文稿。还要特别感谢沙市大学邹时智教授给予的全力支持和帮助。在此一并表示感谢。

由于时间仓促，加上作者学识水平有限，书中难免有错误与疏漏之处，恳请广大读者批评指正。

作　者

内 容 简 介

本书包括MATLAB程序设计语言基础与控制系统MATLAB计算及仿真两大部分：第一部分介绍MATLAB用于控制系统计算及仿真的有关知识，包括MATLAB数值运算基础，M文件与MATLAB函数，MATLAB程序设计基础与MATLAB的文字处理工具Notebook等，并具体介绍了仿真集成环境SIMULINK。第二部分内容涵盖了经典控制理论、电力拖动控制系统、位置随动系统、过程控制系统、现代控制理论等方面，介绍了控制系统仿真基础，自动控制系统的MATLAB计算及仿真，自动控制系统设计，状态空间分析的MATLAB计算及仿真，以及最优控制系统的MATLAB计算及仿真。附录A列出了MATLAB程序设计中经常使用的各种函数命令；附录B列出了MATLAB系统提供的TOOLBOX控制工具箱函数命令。函数的内容极为丰富，都是用于自动控制方面的。

本书示例丰富，大部分示例取材于自动控制理论、电力拖动控制、位置随动控制、生产过程控制、现代控制理论等方面的经典教科书，还有一部分取自实际工程的课题。示例程序都经作者反复调试，既能够在MATLAB里运行，也能够在Notebook中运行（除开一些带鼠标操作的图形函数命令必须在MATLAB命令窗口里运行之外）。

本书可作为自动控制及其相关专业、机械设备与自动化专业或机电一体化专业的学生教材，也可作为自动控制领域科技人员的参考用书。

目　录

第 1 章　控制系统及仿真概述

本章简要概述控制系统计算与仿真的基本知识。本章共 4 节。1.1 节简要介绍自动控制系统的广泛应用，包括电力拖动自动控制系统与工业生产过程控制系统的应用，1.2 节说明控制系统计算机仿真的基本概念，1.3 节介绍控制系统 MATLAB 计算及仿真，通过本章的介绍，使读者了解全书的概貌与本书的主要内容及任务。

1.1　自动控制系统的广泛应用

现代自动控制控制系统应用非常广泛，几乎遍及国民经济的所有部门。本书讨论的控制系统不是广义的、泛指的控制系统，而是工程技术领域里常指的自动控制系统。以控制对象生产工艺过程的性质、特点而论，控制系统大体可分为两大门类：一类是以机械运动为主要生产形式，以电动机为执行机构的“电力拖动自动控制系统”；另一类是以化学反应或者热能转换为主要生产形式，以自动化仪表与装置为检测与执行机构的“工业生产过程控制系统”。

这两类控制系统仅仅是执行机构和系统的设备生产形式不同，其自动控制的理论完全一样，研究的方法以及仿真理论、过程没有什么差别。本书以拖动自动控制系统、过程控制系统为讨论对象，还涉及到了控制系统设计、状态空间分析与最优控制系统等内容。

1.1.1　电力拖动自动控制系统的应用

在现代工业生产中，为了实现各种不同生产工艺过程的要求，需要使用各种各样的生产机械。例如各类生产机床、轧钢机、矿井提升机、球磨机、造纸机、纺织机械、印刷机械、化工机械、电力机车、压缩机、电梯、起重机、卷扬机、挖掘机、榨油机、碾米机、水泵、电动工具乃至家用电器等等，数不胜数。

电动机拖动生产机械运转叫做电力拖动。电力拖动，是电动机将电能转换为机械能的过程，是当代各种生产活动最基本也是最先进的形式。这是新型的电力代替传统而原始的人力或者畜力的历史必然，也是现代文明的标志之一。

作为现代能源主要形式的电能，其生产集中在火力、水力、风力与原子能发电厂进行。电能可远距离输送，简单经济、易于分配、转换与控制，检测方便，价格低廉。电机拖动比其他形式的拖动效率要高得多，原动机与被拖动机械的连接简便。电动机效率高，运转经济。电动机的形式种类与规格很多，具有各种良好的特性，可适应各种不同生产机械的需要。电机拖动易于操作及控制，电动机的起动、制动、反向及调速等控制

简便快捷，调节性能良好。电力拖动可实现远距离控制与自动调节，并进而实现生产过程的自动化。

综上所述，电力拖动自动控制系统已经成为现代工业生产电气化及自动化的基础，而实现工业企业的电气化及自动化对于提高产品质量、改善工人的劳动条件、增加工作可靠性以及提高劳动生产率均有着重大的意义。

1.1.2 工业生产过程自动控制系统的应用

工业生产过程自动控制是指在冶金、机械、石油、化工、电力、轻工、建材、原子能与环保等部门的生产中，对于温度、压力、流量、液位、成分等变量实现的自动控制。它是利用过程检测控制仪表、自动化设备与装置、数字计算机等自动化技术工具，对整个生产过程进行自动检测与控制，以期达到各种最优的技术、经济指标，提高经济效率与劳动生产率、节约能源、改善劳动生产条件、保护生态环境等目的。

工业生产过程自动控制系统（简称过程控制系统），是由控制对象与过程控制仪表所组成。以控制的角度而论，工业生产过程自动控制着重研究串级控制、前馈控制、均匀控制、比值控制、分程控制、选择性控制、多变量控制、多冲量控制以及数字过程控制等方面的课题。

工业生产过程自动控制系统是一门内容极为丰富的综合性应用技术学科，它与工程实践联系紧密，与电力拖动自动控制系统一样，在现代工业生产过程自动化中也得到了十分广泛的应用。

1.2 控制系统计算机仿真的基本概念

1.2.1 系统计算机仿真

控制系统的计算机仿真是一门涉及到控制理论、计算数学与计算机技术的综合性新型学科。这门学科的产生及发展差不多是与计算机的发明及发展同步进行的。计算机仿真是自从1946年世界上第一台电子计算机问世以后，在20世纪70年代初期发展起来的、用来帮助设计人员进行设计的一种新技术。它包含控制系统分析、综合、设计、检验等多方面的计算机处理。计算机仿真基于计算机的高速而精确的运算，以实现各种功能。

系统，是物质世界中相互制约又相互联系着的、以期实现某种目的的一个运动整体，这个整体叫做系统。如果系统用于自动控制，则称之为自动控制系统。

模型，是对所要研究的系统在某些特定方面的抽象。通过模型对原型系统进行研究，将具有更深刻、更集中的特点。模型分为物理模型和数学模型两种。数学模型可分为机理模型、统计模型与混合模型。

系统仿真，就是以系统数学模型为基础，以计算机为工具对系统进行实验研究的一种方法。需特别指出，系统仿真是用模型（即物理模型或数学模型）代替实际系统进行实验和研究，使仿真更具意义。仿真所遵循的基本原则是相似原理，即几何相似、环境相似与性能相似。依据这个原理，仿真可分为物理仿真与数学仿真（即模拟计算机仿真

与数字计算机仿真）。

所谓物理仿真就是应用几何相似原理，仿制一个与实际系统工作原理相同、质地完全相同但是体积小得多的物理模型（例如将飞机模型放在气流场相似的风洞中以模仿真实的飞机在地球的大气中）进行实验研究；所谓数学仿真就是应用性能相似原理，构成数学模型在计算机上进行实验研究。

由于计算机仿真能够为各种实验提供方便、廉价、灵活而可靠的数学模型，因此凡是要用模型进行实验的，几乎都可以用计算机仿真来研究被仿真系统的工作特点、选择最佳参数和设计最合理的系统方案。

随着计算机技术的发展，计算机仿真越来越多地取代纯物理仿真。现在所称谓的仿真，主要是指计算机参与的计算机仿真。计算机仿真是一门综合性的新学科，它既取决于计算机工具本身硬件与软件的发展，又依赖于仿真计算方法在精度与效率方面的研究与提高，还要服从于计算机仿真对象学科领域的发展需要。所以计算机仿真是多种学科互相渗透、相互融合又与多种学科相关联的边缘科学。

而自动控制系统的计算机仿真，它是一门涉及到计算机技术、计算数学与控制理论、系统辨识、控制工程以及系统科学的综合性学科。它为控制系统的分析、计算、研究、综合设计以及自动控制的计算机辅助教学提供了快速、经济、科学及有效的手段。

1.2.2 控制系统计算机仿真的过程

控制系统仿真，就是以控制系统的模型为基础，主要用数学模型代替实际的控制系统，以计算机为工具，对控制系统进行实验和研究的一种方法。

通常，控制系统仿真的过程按以下步骤进行：

（1）建立自控系统的数学模型

系统的数学模型，是描述系统输入、输出变量以及内部各变量之间关系的数学表达式。描述自控系统诸变量间静态关系的数学表达式，称为静态模型；描述自控系统诸变量间动态关系的数学表达式，称为动态模型。最常用的基本的数学模型是微分方程与差分方程。

根据系统的实际结构与系统各变量之间所遵循的物理、化学基本定律，例如牛顿定律、克希霍夫定律、运动动力学定律、焦耳—楞次定律等等来列写出变量间的数学表达式以建立数学模型。这是解析法建立数学模型。

对于很多复杂的系统，则必须通过实验方法并利用系统辨识技术，考虑计算所要求的精度，略去一些次要因素，使模型既能准确地反映系统的动态本质，又能简化分析计算的工作。这是实验法建立数学模型。

控制系统的数学模型是系统仿真的主要依据。

（2）建立自控系统的仿真模型

原始的自控系统的数学模型比如微分方程，并不能用来直接对系统进行仿真，还得将其转换为能够对系统进行仿真的模型。

对于连续系统而言，将像微分方程这样的原始数学模型，在零初始条件下进行拉普拉斯变换，求得自控系统传递函数之数学模型。以传递函数模型为基础，等效变换为状态空间模型，或者将其图形化为动态结构图模型，这些模型都是自控系统的仿真模型。

对于离散系统而言，有像差分方程这样的原始数学模型以及类似连续系统的各种模型，这些模型都可以对离散系统直接进行仿真。

（3）编制自控系统仿真程序

对于非实时系统的仿真，可以用一般的高级语言，例如 Basic 或 Fortran 或 C 等语言编制仿真程序。对于快速的实时系统的仿真，往往用汇编语言编制仿真程序。当然也可以直接利用仿真语言。

应用 MATLAB 的 TOOLBOX 工具箱及其 SIMUKINK 仿真集成环境作仿真工具，这就是 MATLAB 仿真。控制系统的 MATLAB 仿真是控制系统计算机仿真一个特殊软件工具的子集。

（4）进行仿真实验并输出仿真结果

通过仿真实验对仿真模型与仿真程序进行检验和修改，再按照系统仿真的要求输出仿真结果。

1.3 控制系统 MATLAB 计算及仿真

1.3.1 先进的软件 MATLAB

MATLAB 程序设计语言是美国 Mathworks 公司 20 世纪 80 年代中期推出的高性能数值计算软件。Mathworks 公司经过十几年的开发、扩充与不断完善，使 MATLAB 已经发展成为适合多学科、功能特强、特全的大型系统软件。在国外 MATLAB 已经经受了多年考验。在欧美高校，MATLAB 已经成为线性代数、自动控制理论、数理统计、数字信号分析与处理、动态系统仿真等高级课程的基本数学工具，成为高校大学生、硕士生、博士生必修课。

在科学研究与工程技术应用中常常要进行大量的数学运算。在当今计算机时代，通常的作法是借助高级语言 Basic、Fortran 和 C 语言编制计算程序，输入计算机做近似计算。但是，这需要熟练地掌握所用语言的语法规则与编制程序的相关规定，而且编制程序绝非易事。

当初的MATLAB是Cleve Moler 博士于1967年用 Fortran 编写的(第二代的MATLAB的主要部分是用C语言编写的）。1980年前后，Moler开始筹建 MATLAB。Moler开发MATLAB的最初目的只是为线性代数的矩阵运算提供一个运算工具，但后来功能逐渐完善扩充，可以进行各种科学与工程技术计算，应用范围也越来越广，而且 MATLAB 的函数命令简单高效，特别容易学习和方便的使用。于是，Moler 博士等一批数学家与软件专家组建了 Mathworks 软件开发公司，专门研究、扩展并改进MATLAB，该公司于 1992年推出了 MATLAB 4.0 版本，并于 1993 年推出其微机版，1994 年又推出了 MATLAB 4.2 扩充版，1998 年的 MATLAB 5.1 版本，近两年又相继推出 MATLAB 的 5.2 与 5.3 版本。

MATLAB 有以下主要特点：

（1）功能强大、适用范围广

MATLAB 可用于线性代数里的向量、数组、矩阵运算，复数运算，高次方程求根，

插值与数值微商运算，数值积分运算，常微分方程的数值积分运算、数值逼近、最优化方法等，即几乎所有科学研究与工程技术应用需要的各方面的计算，均可用 MATLAB 来解决。

现有的资料表明，这些运算，在自动控制、生物医学工程、语音处理、图像信号处理、雷达工程、信号分析、计算机技术、汽车制造、建筑业、航空航天、分子模型研究以及半导体制造业等各行各业都有极广泛的应用。可以说，无论从事工程技术方面的哪个学科，在 MATLAB 里都能有相应合适的功能解决该学科的计算问题。

（2）编程效率高

MATLAB 程序设计语言提供了丰富的库函数（称为 M 文件），既有常用的基本库函数，又有种类齐全、功能丰富多样的专用库函数（工具箱 TOOLBOX 就是专门功能库函数）。函数即是预先编制好的子程序。在为进行数学运算编制程序时，这些库函数都可以被直接调用，而不必将其子程序的命令或语句再逐一列出。无疑，这会大大提高编程效率。

MATLAB 的基本数据编程单元是不需要指定维数的复数矩阵，所以在 MATLAB 环境下，数组（向量或矩阵）的操作如同数的操作一样简单方便，不必像其他 Basic、Fortran 和 C 等高级语言，因为基本数据单元是数而要事先定义数组，然后才能进行有关操作。这样的编程效率肯定不如 MATLAB 的高。

在科学与工程应用的数值计算领域里，有文献指明，使用 MATLAB 语言的程序设计比使用 Basic 、Fortran 和 C 等语言进行程序设计的编程效率要高几倍。

（3）界面友好、用户使用方便

首先，MATLAB 具有友好的用户界面与易学易用的帮助系统。MATLAB 的函数命令繁多，功能各异。用户在命令窗里通过 help 命令可以查询某个函数的功能及用法，命令的格式极为简单（格式为：help 命令或函数）。用户在命令窗里通过某些命令还可以查询某个函数的路径以及查询某个子目录中的函数集合。这样，面对 MATLAB 的强大功能与各种先进技术，即便是初学者，也不会望而生畏。因为 MATLAB 已为用户提供了学习它的方便之路。

其次，MATLAB 程序设计语言把编辑、编译、连接、执行、调试等多个步骤融为一体。如果直接在命令行输入语句（命令），包括调用 M 文件的语句，每输入一条（语句），就立即完成编译、连接和运行的全过程。如果将 MATLAB 源程序编辑为 M 文件，编辑后的源文件就可像库函数一样直接运行，而不再需要编译和连接。运行 M 文件时，如果有错，计算机屏幕上会给出详细的出错信息提示，让用户修改，直到正确为止。

再者，MATLAB 语言可设置中断点，存储多个中间结果。除此以外，它还可进行跟踪调试。MATLAB 语言灵活方便，其调试手段丰富，调试速度快。

还有，在 MATLAB 里，既可执行程序（即 M 文件），又可通过人机对话，调用不同的库函数（即子程序），方便快速地达到用户自己的目的，以实现 MATLAB 的交互功能。

最后，MATLAB 是演算纸式的科学工程计算语言。 使用 MATLAB 编程运算与人进行科学计算的思路和表达方式完全一样，MATLAB 的语法更贴近人的思维方式，用 MATLAB 编写程序，犹如在一张演算纸上排列书写公式，运算、求解问题十分方便。

（4）扩充能力强

MATLAB 程序设计语言不仅为用户提供了可直接调用的丰富的库函数（即 M 文件），而且在 MATLAB 语言环境下，用户还可以根据需要，自行建立或扩充完成指定功能的 M 文件（即新的库函数），与 MATLAB 提供的系统里的库函数一样保存，同样使用，以提高 MATLAB 使用效率，并丰富、扩充它的功能。

另外，为了充分利用 Basic 、Fortran 和 C 等语言的资源，包括用户已经编写好的 Basic、Fortran 和 C 语言程序，通过建立 Mex 文件的形式，进行混合编程，能够方便地调用 Basic、Fortran 和 C 语言的子程序，以进一步丰富及扩充 MATLAB 程序设计语言的功能。

（5）语句简单、内涵丰富

MATLAB 最基本的语句结构是赋值语句，语句的一般形式为

变量名列表 = 表达式

其中等号左边的变量名列表为 MATLAB 的语句返回值，等号右边是表达式的定义，它可以是 MATLAB 允许的矩阵运算，也可以是 MATLAB 的函数调用。

MATLAB 程序设计语言最重要的成分是函数。函数调用的一般形式为

[a,b,c,……] = fun （d,e,f,……）

即一个函数由函数名、输入变量 d,e,f,……与输出变量 a,b,c,……组成，同一函数名 F，不同数目的输入变量及不同数目的输出变量，代表着不同的含义。这不仅使 MATLAB 的库函数功能更加丰富，而且大大减少了需要的磁盘空间，使得 MATLAB 编写的 M 文件简单、精炼而高效。

（6）强大方便的图形功能

MATLAB 提供了许多“高级”图形函数，可以绘制出多姿多彩的图形。例如，绘制二维、三维曲线并对平面或空间多边形填充；绘制三维曲面并对其进行复杂操作：通过开关参数实现曲面的透明或消隐；利用缩放功能对图形的局部位置取景放大；调整观察角与方位角考察空间曲面的不同侧面；用光照效果对曲面进行明暗处理以增强其立体感；为渲染曲面的空间特性而在网线间填色等等。

MATLAB 还开发了一些面向图形对象的“低级”图形函数，可以访问硬件系统建立各种“低级”图形对象，它们以图形句柄为界面。用户使用图形句柄可以操作图形的局部元素。

MATLAB 有一系列绘图函数（命令），适用于不同的坐标体系，例如：线性坐标、对数坐标、半对数坐标、极坐标及三维坐标，只需调用不同的绘图函数（命令），即可在图上标出图形的标题、X、Y 轴的标注。格（栅）绘制也只需调用相应的命令，简单易行。

另外，在调用绘图函数时，调整自变量可绘出不同的线形：点线、实线、复线或多重线；调整绘图函数的另一些自变量可绘出不同颜色的各种线条。这就使得在 MATLAB 环境下绘制的图形尤其清晰、精美、绚丽多彩。利用 MATLAB 提供的这些图形技术可以创造出无与伦比的彩色世界。

（7）MATLAB 的活笔记本功能

MATLAB 的 Notebook 成功地把 Microsift Word 与 MATLAB 集成为一个整体，为文

字处理、科学计算、工程设计构造了一个完美统一的工作环境，Notebook 是一个能够解决各种计算问题的文字处理软件。只要在命令窗口中执行 Notebook 或者在 Word 环境中建立 M-book 模板，就可以进入一个新环境。在编辑科技文稿的同时可进行科学演算（数值的或者符号的），还可以作图。这些演算的结果可以即时显示于操作命令之后。在这个环境中输入的一切命令能够被随时激活、修改、重新运算并更新原有结果。故 Notebook 称为 MATLAB 的“活”笔记本。这对于撰写科技论文的工程技人员，对于编写理工学科教材的教师，对于演算理工学科习题的广大学生，MATLAB 的 Notebook 确实是一个极为理想的工具。

（8）功能齐备的自动控制软件工具包

Mathworks 软件公司先后开发了许多工具包，世界上很多从事自动控制的知名专家，如英国、美国、瑞典、日本等国学者在自己擅长的领域里编写了具有特殊功能的工具包：诸如控制系统工具箱（Control Systems Toolbox）；信号处理工具箱（Signal Processing Toolbox）；系统识别工具箱（System Identification Toolbox）；鲁棒控制工具箱（Robust Control Toolbox）；μ分析与综合工具箱（μ-Analysis and Synthesis Toolbox）；定量反馈理论工具箱（QFT Toolbox）；神经网络工具箱（Neural Network Toolbox）；多变量频域设计工具箱（Multivariable Frequency Design Toolbox）；最优化控制工具箱（Optimisation Toolbox）等等，还在不断地扩充、丰富与完善。每一个工具箱都是该控制领域里的最权威、最先进的计算与仿真软件程序。这些工具包都已成为 MATLAB 的重要组成部分。也使 MATLAB 成为自动控制最先进的强有力工具。

MATLAB 及其工具包的开发，使得它在科学计算、工程技术与自动控制方面的应用愈来愈普遍。

MATLAB 语言易学易用，不要求使用者有高深的数学与程序语言的知识，不需要使用者深刻了解算法与编程技巧。在诸多领域里，MATLAB 有一般高级语言难以比拟的优势，无论是作为科学研究与工程运算的工具，还是作为计算机辅助的教学工具，都是不可多得的。所以，自推出后不久 MATLAB 即风行美国、西欧，流传世界。

1.3.2 控制系统 MATLAB 计算及仿真的优秀性能

由于 MATLAB 以上的特点，所以它特别适合用来对控制系统进行计算与仿真。

首先，MATALB 运算功能强大，它提供的向量、数组、矩阵运算，复数运算，求解高次方程，常微分方程的数值积分运算、最优化方法等等，这些都是在控制理论及控制系统里经常遇到的计算课题。MATLAB 就这一点已与自动控制密切联系在一起。

其次，由于世界上从事自动控制的多个知名专家，在自己擅长的控制领域里开发了具有特殊功能的 TOOLBOX 工具箱，使得 MATLAB 从一个数值运算工具变成为自动控制计算与仿真的强有力工具。MATLAB 的控制箱里，软件内容丰富，系统门类齐全，已覆盖了控制系统的各个领域，每一个工具箱都是当今世界上该控制领域里的最权威、最顶尖的计算与仿真程序软件。

再有，MATLAB 内涵丰富，扩充能力强，编程效率特高。不仅 MATLAB 的开发者可以编制软件程序，使用者同样可以为实现新功能或特殊功能开发编制软件程序，并将其放到 MATLAB 里去，使 MATLAB 的功能不断扩充逐步完善。

还有，MATLAB 语言语句简单，极其容易学习与使用。自动控制本身就有很多理论问题、系统实现问题需要研究，再要为学习这种语言及其语法规花太多的时间与精力是不可取的。MATALB 正好具有语言简单、掌握方便的特点，是一个理想的工具。

最后，MATLAB 界面友好，使得从事自动控制的科技工作者乐于接触它，愿意使用它。MATLAB 的强大方便的图形功能，可以使得重复、繁琐计算与绘制图形的笨重劳动被简单、轻而易举的计算操作所代替。而且数据计算准确，图形绘制精密，这是过去从事本专业的人所追求与期盼的事情。

随着 MATLAB 软件的出现，它的 TOOLBOX 工具箱与 SIMULINK 仿真工具，为控制系统的计算与仿真提供了一个强有力的工具，使控制系统的计算与仿真的传统方法发生了革命性的变化。MATLAB 已经成为国际、国内控制领域内最流行的计算与仿真软件。

第 2 章　MATLAB 程序设计语言

现在流行的 MATLAB 5.3 经历了十多年的补充、完善与几个版本的升级换代，已经发展成为一个内容极为丰富的庞大系统。MATLAB 软件及其 TOOLBOX 工具箱与 SIMULINK 仿真工具，为控制系统的计算与仿真提供了一个强有力的工具。为此，有必要扼要介绍将 MATLAB 用于控制系统计算仿真的有关内容。本章共 6 节：2.1 节首先对 MATLAB 软件的安装与启动作简要介绍；2.2 节介绍 MATLAB 的各种环境界面；2.3 节介绍 MATLAB 的各种数值运算；2.4 节介绍 M 文件与 MATLAB 函数；2.5 节介绍 MATLAB 程序设计的基础知识；2.6 节介绍 MATLAB 的文字处理工具 Notebook。

2.1　MATLAB 安装与启动

本节主要介绍 MATLAB 5.3 的安装与启动及其进入其环境。MATLAB 5.3 的安装虽说不是太复杂，但是如果有一个安装指南，就可按图索骥，保证一次成功。安装成功后，启动就很简单了。启动 MATLAB 5.3 后，就会看到它精彩的界面以及琳琅满目的内容。于是读者便会逐渐进入 MATLAB 系统并逐渐熟悉、掌握 MATLAB。

2.1.1　MATLAB 5.3 的安装

MATLAB 5.3 比 MATLAB 的老版本提供了许多更强的新功能和更全面、更方便的联机帮助信息，当然，MATLAB 5.3 也比以前的版本对于软、硬件提出了更高的要求。下面介绍 MATLAB 5.3 在 PC 机上安装的资源要求和安装过程。

MATLAB 5.3 可以安装到 IBM 或与之完全兼容的带数学协处理器的 Intel 486、Pentium 及其以上的 PC 机上。

1. MATLAB 5.3 对 PC 机系统的资源要求

（1）Microsoft Windows 95（或 98 或 2000）或 Microsoft Windows NT 的操作系统。

（2）推荐使用八倍速以上的光驱。

（3）至少 8 位或以上显示器适配卡和至少能显示 256 色的彩色显示器。

（4）推荐使用 32MB 以上内存储器。

（5）对于网络用户，只支持分配了静态 IP 地址的用户。

（6）安装声卡。

（7）如果用户想安装 Notebook，计算机里需先安装 Microsoft Word 6.0 、Microsoft Word 7.0、Microsoft Word 97 或者 Microsoft Word 2000 中的任何一种。顺便说明，本书的

文稿就是在 Microsoft Word 2000 中编辑的。

（8）如果用户想创建 MEX 文件，需安装 Microsoft、Borland 或 Watcom 的 C 语言编辑器，或者是 Microsoft 的 Power Station Fortran 编辑器。

（9）如果用户想完全阅读 MATLAB5.3 的帮助信息，用户需预先安装 Netscape Navigator 2.0 以上版本或者是 Microsoft Internet Explorer 3.0 以上版本来阅读其中的超文本帮助信息，用户还需安装 Adobe Acrobat Reader 来阅读其中的 PDF 格式的帮助文件。在 MATLAB 5.3 的安装盘上，附带了 Adobe Acrobat Reader。

2. MATLAB 5.3 的安装过程

现在流行使用的较新版本是 MATLAB 5.3，目前同时使用的还有较早版本 5.0 版、5.1 版与 5.2 版等。本书要介绍的是 MATLAB 5.3 版，在此简述它的安装过程。

（1）启动计算机后，将 ATLAB 5.3 的光盘放入光驱中，如果操作系统为 Windows 95（或 Windows98 或 Windows2000），会立即自动运行安装程序。也可以在“我的电脑”中或直接进入光驱后再双击 MATLAB 的 setup.exe 文件，显示如图 2-1 所示的 MATLAB 封页界面，开始启动安装程序。

（2）安装程序会自动打开 MATLAB 安装对话框（图 2-2），按照安装程序的提示，可进入到用户输入信息的界面。用户在相应的位置输入用户名称、公司名及产品注册码，并选择要安装的 MATLAB 的版本，按下【Next】按钮即可继续安装。



图 2-1 MATLAB 封页界面

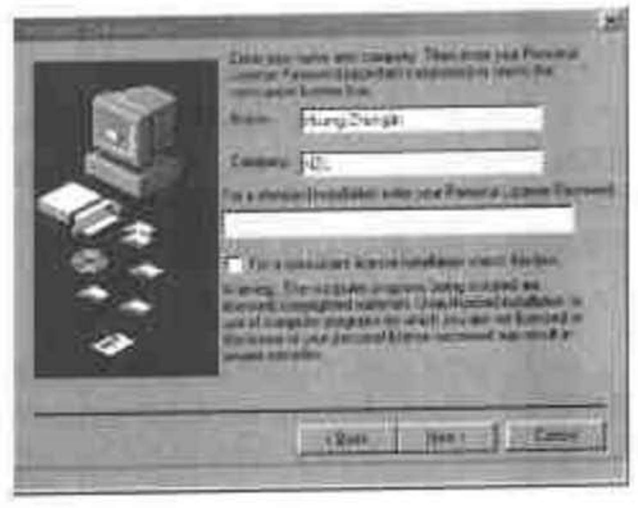


图 2-2 MATLAB 安装对话框

（3）安装程序会自动打开 MATLAB 安装内容选择对话框（图 2-3），MATLAB 的组件、安装目录以及安装所需要的磁盘空间等信息均显示在图中。按照安装程序的提示，可以通过选中组件前对应的复选框来决定要安装的组件。请特别注意安装光盘时提示不允许安装的组件不能选中，否则安装无效。在图 2-3 中，还显示安装 MATLAB 时所需要的磁盘空间以及用户计算机的硬盘可用空间。如果可用磁盘空间不够，可以减少要安装的组件或改换使 MATLAB 能够安装完的磁盘。单击【Browse】按钮，可以浏缆 MATLAB 的安装目录（图 2-4），也可在图 2-4 中的【Path】框中改变安装路径。按下【OK】按钮即可回到图 2-3 的对话框，再按下【Next】按钮即可继续安装。

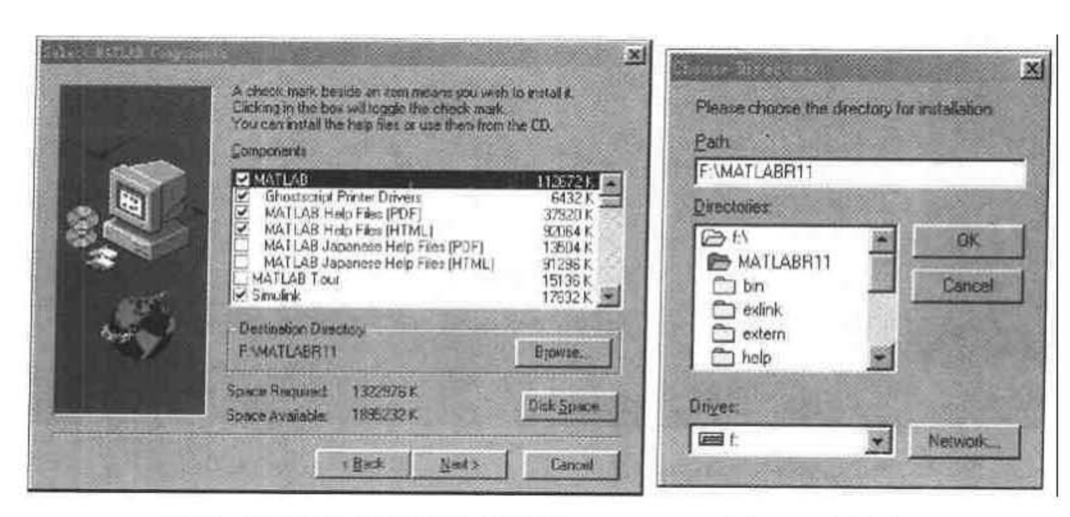


图 2-3　MATLAB 安装组件选择对话框　　　　图 2-4　安装路径对话框

（4）如图 2-5 所示，开始 MATLAB 的正式安装。安装 MATLAB 5.3 须要持续大约 20 分钟左右，安装过程中，如果计算机上原来已经安装过 MATLAB 系统，安装程序会使屏幕弹出如图 2-6 的提示对话框，供选择对安装过 MATLAB 的覆盖（重写）情况。例如用户可以选中“Overwrite all components”，将从前的内容全部覆盖。再按下【Next】按钮，继续进行下一步安装。

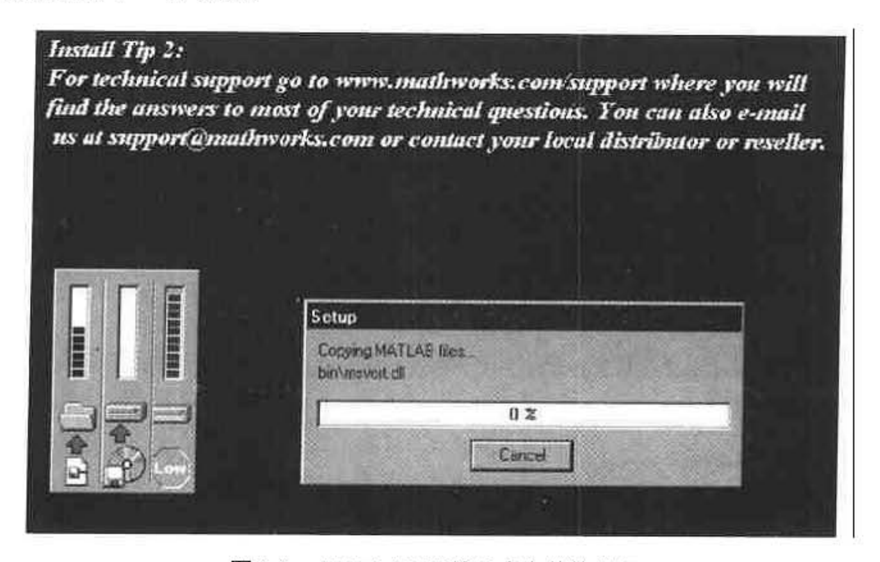


图 2-5　MATLAB 开始正式安装的界面

（5）安装完成时，有 MATLAB 安装结束的提示界面如图 2-7 所示。在图中可以选择“立即重新启动计算机”或“不立即重新启动计算机”，然后再按下【Finish】按钮，则安装过程即告结束。同时在 Windows 的操作桌面上，生成 MATLAB 快捷图标。

需要特别说明的是，MATLAB5.3 与 Word 组合成的中文 Notebook 字处理软件环境，作为 MATLAB 的一个组成部分，是需要另外特别安装的，并且与 MATLAB 所有老版本的 Notebook 的安装都不相同。

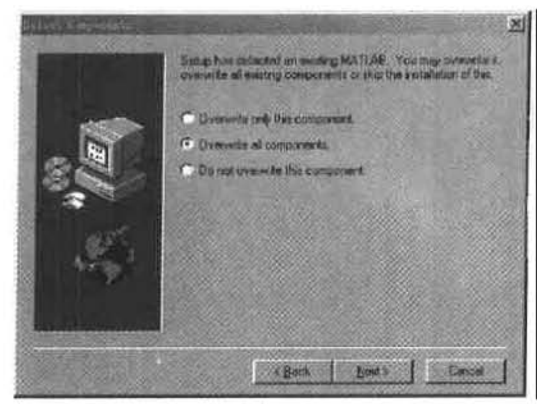


图 2-6　覆盖已安装组件选择对话框



图 2-7　安装结束界面

（6）Notebook 的安装。MATLAB Notebook 所需要的系统环境是两个系统软件：

①Microsoft Word 6.0（或 7. 0）或者 Microsoft Word 97 （或 2000）。

②MATLAB 5.x。

如果安装的是 MATLAB 5.1 或 5.2 版本，Notebook 的安装是在 MATALB 的安装过程中进行的。即在其安装过程进行到 MATLAB 主体和工具箱安装完毕后，安装程序将询问用户是否安装 Notebook，只需按照提示步骤一步一步输入信息即可完成 Notebook 的安装。

本书要介绍 MATLAB 5.3 的 Notebook 安装过程。与以上介绍的不同，Notebook 的安装是在 MATALB 5.3 的安装完成后进行的。在 MATLAB 5.3 系统中有一个 notebook.m 函数，Notebook 的安装就是借助它实现的。Notebook 安装的具体步骤如下：

①启动 MATLAB 5.3，打开 MATLAB 命令窗口。

②在 MATLAB 命令窗口中运行函数命令“notebook -setup”（请注意 notebook 后有空格与‘-’号），按回车键后，有如图 2-8 的提示。

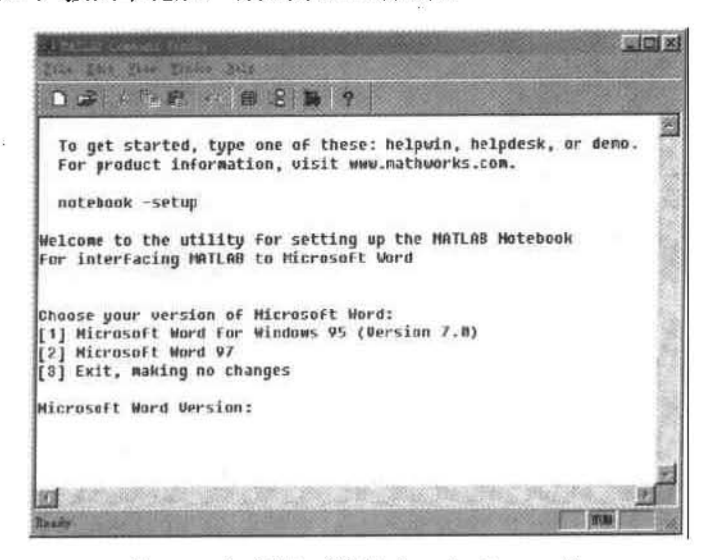


图 2-8　命令窗口中运行“notebook -setup”

③在“Microsoft Word Version：”后，回答其版本号。如果用户计算机上安装的是 Microsoft Word 97 或者是 Microsoft Word 2000，回答时均可输入不带括号的“2”，按回车键后，则出现如下提示：

```
You will be presented with a dialog box.  Please use it to select your copy of the
Microsoft Word 97 executable (winword.exe).
Press any key to continue...
```

按任意键，即可弹出如图 2-9 所示的对话框，供用户查询并找出 Winword.exe 文件所在的路径，当确认后，即选中“Winword.exe”，并按下【打开】按钮。此时，MATLAB 命令窗口即给出安装结束的提示：

```
Notebook setup is complete.
```



图 2-9　用户确认“Winword.exe”文件所在路径

2.1.2　MATLAB 5.3 的启动

当 MATLAB 及其 Notebook 成功安装完毕后，即可运行 MATLAB。MATLAB 5.3 的启动有三种方法。

（1）MATLAB 5.3 的图标快捷启动方式

在 Windows 98（或 2000）的操作桌面上，双击其图标即可启动 MATLAB。

（2）MATLAB 5.3 的菜单启动方式

在 Windows 98（或 2000）的操作桌面上，点击【开始】按钮，选择【程序】选项，从其下拉菜单中再选择【MATLAB】选项，从其下拉菜单中又再点击【MATLAB5.3】选项，即可启动 MATLAB（图 2-10）。

（3）MATLAB 5.3 随 Word 文件打开启动方式

在已经启用 M-book 模板，即主菜单栏含有菜单项【Notebook】的 Word 窗口里（参看图 2-11），若是已经有用户开发或编辑过的 Word 文件，那么在打开该文件时，就相应地启动了 MATLAB。

2.1.3　Notebook 的启动

所谓启动 Notebook，就是指从不带【Notebook】主菜单的 Word 常规窗口变成已启用 M-book.dot 模板的 Word 窗口，此时 Word 窗口主菜单已增添了【Notebook】菜单项。

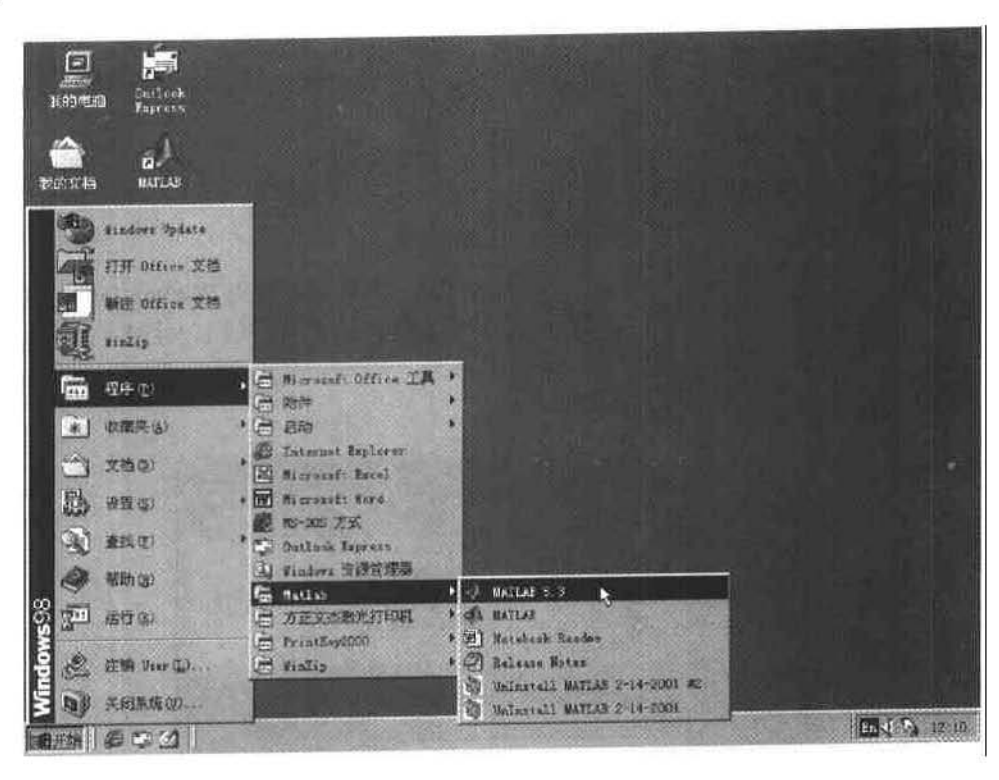


图 2-10 菜单方式启动 MATLAB 5.3

Notebook 有两种启动方法：一种是从 MATLAB 中启动 Notebook，另一种是从 Word 中启动。

第一种方法很简单。在 MATLAB 命令窗口里，输入 notebook 命令，回车后（即运行该命令），即有如图 2-11 所示的 Word 界面，主菜单上有【Notebook】菜单项，说明启动成功。

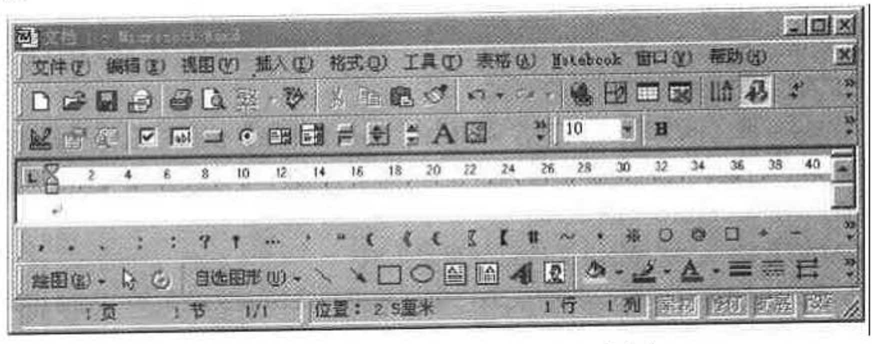


图 2-11 Notebook 启动成功的 Word 界面

第二种方法稍微复杂些。在 Windows 下启动 Word，从主菜单的【文件】菜单项的下拉菜单中选择【新建】子项，即有图 2-12 所示的对话框，选中“M-book.dot”图标，按下【确定】按钮，同样有图 2-11 所示的 Notebook 启动成功的 Word 界面。

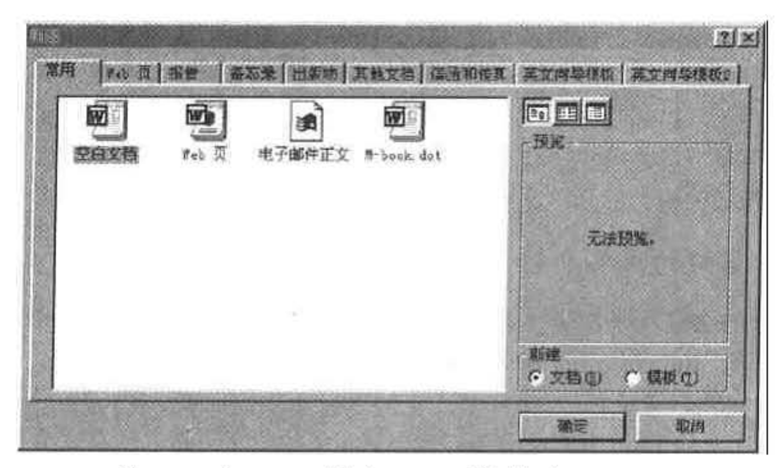


图 2-12　在 Word 下创建 M-book 模板启动 Notebook

2.2　MATLAB 环境

MATLAB 既是一种高级计算机语言，又是一个编程环境。作为编程环境的 MATLAB 提供了很多方便用户管理变量、输入输出数据以及生成、管理 M 文件的工具。本节要介绍 MATLAB 的命令窗口及其帮助菜单、工作空间、变量浏览器、程序编辑/调试器、路径浏览器与 Work 子目录等。

2.2.1　MATLAB 的命令窗口

MATLAB 的命令窗口是 MATLAB 的重要组成部分，MATLAB 系统在计算机里的存在，主要是以命令窗口形式体现的。在 MATLAB 命令窗口里，使用者输入命令、程序、点击菜单项或工具栏按钮，指挥 MATLAB 计算、仿真，其结果也都在命令窗口显示。所以命令窗口是使用者和 MATLAB 交互的地方。了解窗口各部分内容是很有必要的。

正如 2.1.2 节介绍的要打开 MATLAB，在 Windows 环境下，如果已经在 Windows 的桌面上建立了快捷方式图标，双击 MATLAB 图标或者点击【开始】按钮的菜单里的 MATLAB 选项命令，都能打开如图 2-13 所示的命令窗口。

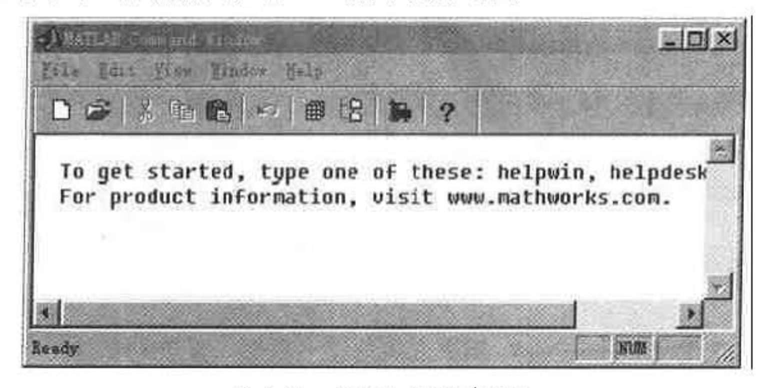


图 2-13　MATLAB 命令窗口

在图 2-13 中，最上面标题栏显示有“MATLAB Command Window”，标题栏的右边从左到右依次为窗口最小化按钮、窗口缩放按钮和窗口关闭按钮。标题栏下面的主菜单包含有【File】（文件）、【Edit】（编辑）、【View】（查看）、【Window】（窗口）、【Help】（帮助）五项菜单项。菜单栏下面的工具栏显示有 10 个工具按钮。工具栏下面是命令编辑区，它占了窗口的绝大部分。以下介绍命令窗口的菜单命令、工具栏和命令编辑区。

2.2.1.1 命令窗口的菜单命令

MATLAB 的菜单项即是菜单命令。由图 2-13 看出，命令窗口的主菜单由 File、Edit、View、Windows、Help5 组菜单命令组成。

1. File（文件）菜单

点击菜单栏上的【File】菜单命令，就会显示下拉式子菜单，现在来介绍各子菜单项命令的作用。

（1）【New】命令

执行【New】命令,下拉菜单有三项子菜单命令：【M-file】、【Figure】和【Model】。

第一项【M-file】命令用于打开 MATLAB 的 M 文件编辑/调试器，并自动打开一个空白 M 文件；通过 M 文件编辑/调试器，用户可以新建或编辑已有的 M 文件，还可调试 MATLAB 程序。第二项【Figure】命令用来建立一个空白图形窗口。MATLAB5.3 的图形窗口（图 2-14），比从前的版本增加了许多新功能，集中反映在【Tools】菜单项里。【Tools】菜单命令下拉后有 11 项子菜单命令，现分述如下。

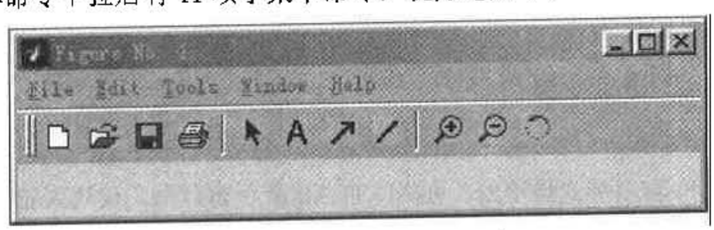


图 2-14　MATLAB5.3 的图形窗口

【Show Toolbar】用来显示或隐藏工具栏。

【Enable Plot Editing】用来允许是否对窗口内的图形进行编辑（等效于按下图形工具栏中的▒按钮）。

【Axes Properties】其对话框用来设置坐标轴属性。

【Line Properties 】其对话框用来设置图形中的线条属性。

【Text Properties】其对话框用来设置图形中的文本属性。

【Unlock Axes Position】用于在图形窗口内自由拖动被选中的坐标轴。

【Show Legend】与【Hide Legend】用于显示或者隐藏图例。单击【Show Legend】时，将显示图例，此时该命令改变为【Hide Legend】。再次单击该命令时，将隐藏图例。

【Add】用来在图形上添加文本、箭头及直线（等效于分别按下图形工具栏中的▒、▒、▒按钮）。

【Zoom In】与【Zoom Out】用来放大与缩小图形（等效于分别按下图形工具栏中的▒、▒按钮）。

【Rotate 3D】将图形放入三维体系，且可任意旋转该三维坐标系（等效于按下图形工具栏中▩的按钮）。

第三项【Model】命令用于自动打开 SIMULINK 仿真库（图 2-15），并打开一个用于建立新模型的窗口（图 2-16）。

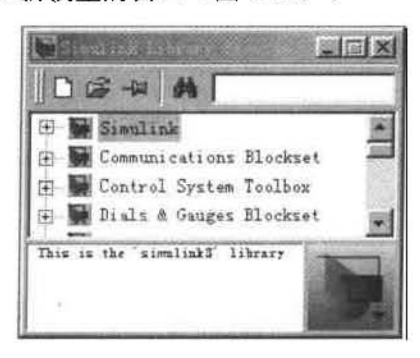


图 2-15　SIMULINK 仿真器

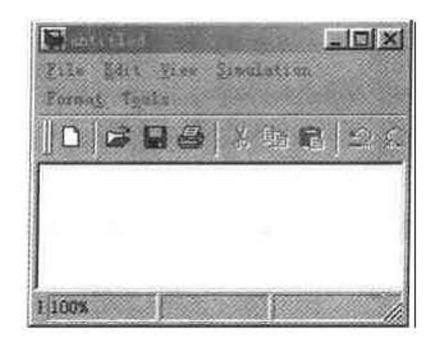


图 2-16　新模型的窗口

（2）【Open】命令

执行【Open】命令时，MATLAB 将打开【Open】对话框。用户可以在对话框中搜寻要打开的 MATLAB 文件所在的路径，进而选中文件并将其打开。

（3）【Open Selection】命令

在 MATLAB 命令窗口的命令编辑区，选中一个 M 文件的文件名，再使用【Open Selection】命令，将用 M 文件编辑/调试器打开该 M 文件。

（4）【Run Script】命令

点击【Run Script】选项命令，将会弹出一个对话框，对话框提示用户输入脚本文件的存储路径和文件名，当确认后，将执行该脚本文件。

（5）【Load Workspace 】命令

MATLAB 为用户提供的加载与保存命令，使用户能够随时保存 MATLAB 工作的内容以及随时把保存的信息重新加载到 MATLAB 的工作空间内。执行该命令将打开一个【Load Workspace 】对话框，在框中列出了【MATLABR11\Work】路径下被保存的后缀为.mat 的文件。

（6）【Save Workspace As】命令

【Save Workspace As】命令用来保存 MATLAB 文件，该命令使用二进制的 MAT 文件保存工作空间的内容。执行该项命令将打开一个对话框【Save Workspace As】，在该框中，有被保存文件的【MATLABR11\work】存储路径和子目录，即以后缀为.mat 的命名文件对当前工作空间的内容进行保存。

（7）【Show Workspace】命令

使用该命令将打开 MATLAB 工作空间浏览器。MATLAB 工作空间浏览器中，显示当前工作空间中所有变量的类型 、大小及占用存储空间等信息。

（8）【Show Graphics Property Editor】命令

执行此命令选项将打开图形属性编辑器（图 2-17）。在图形属性编辑器里，可以选择 MATLAB 已经打开的图形窗口，对每个图形对象的属性值进行修改。

（9）【Show GUI Layout Tool】命令

执行该选项命令，MATLAB将会打开图形界面设计向导（图2-18）。在图形界面设计向导里，用户可以通过选中设计向导里的图形模板，然后在打开的图形窗口里进行图形绘制，以生成精美的图形画面。

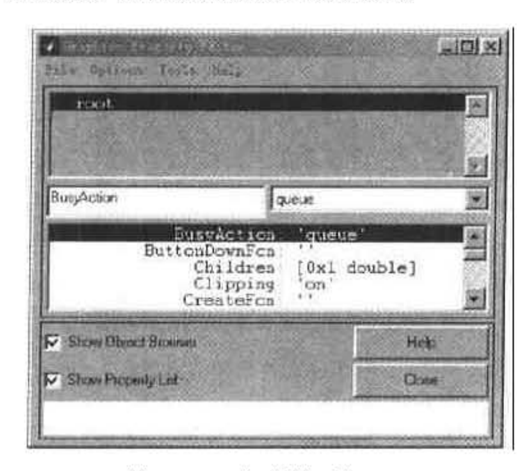


图2-17　图形属性编辑器

图2-18　图形界面设计向导

（10）【Set Path】命令

执行该命令，将打开MATLAB的路径浏览器（图2-19）。通过路径浏览器可以搜索MATLAB所有的M文件的路径，而搜索路径就是指M文件目录的有序列表。

（11）【Preferences】命令

使用该命令，将打开一个参数设置对话框（图2-20），该对话框为重叠的3张选项卡，供用户设置MATLAB工作环境的外观与操作的相关属性。

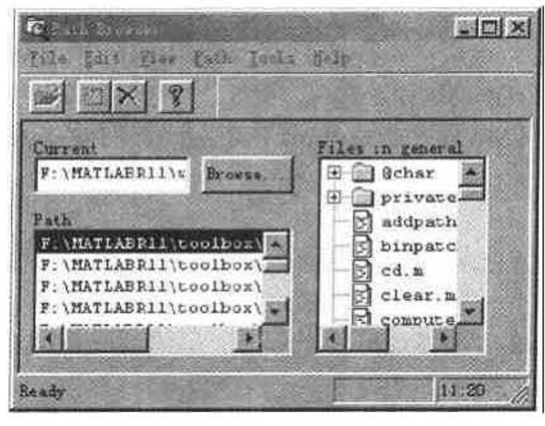


图2-19　MATLAB的路径浏览器

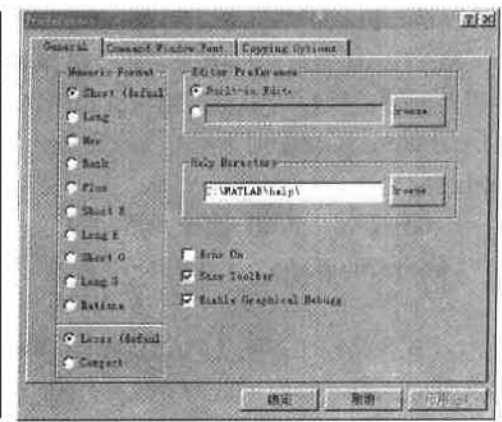


图2-20　【General】选项卡

设置参数是通过设置这三张格式选项卡来实现的。

①第一张【General】选项卡

【General】选项卡（图 2-20 ）主要用于控制命令编辑区数据的显示格式，选择程序编辑器，指定帮助目录，命令执行时是否回显，命令与命令之间以及变量名和变量的

值之间是否留空行等。

在【General】选项卡上左边的【Numeric Format】框中，用户可以用 11 个单选钮来设置 MATLAB 命令窗口显示的数据格式。

MATLAB 提供的数据格式如表 2-1 所示。

表 2-1 命令窗口的数据格式

数据格式名称	数据格式含义	数据格式名称	数据格式含义
Short	短格式	Long E	长格式指数形式
Long	长格式	Short G	短格式紧缩形式
Hex	十六进格式	Long G	长格式紧缩形式
Bank	金融格式	Rational	有理数格式
Plus	+格式	Loose	稀疏格式
Short E	短格式指数形式		

在【General】选项卡（图 2-20）的左下部分，有【Loose（default）】和【Compact】两个单选按钮用来控制命令窗口中数据的输出风格。当选中【Loose（default）】项时，在命令窗口中的显示结果中加入一些空行；而选中【Compact】项时，则压缩掉输出到命令窗口中的许多空行。

在【Editor Preference】框中，当选中【Built-in Editor】项时，则使用 MATLAB 内置的编辑/调试器编写和调试程序；若选中【Browse】项，允许用户选择自己惯用的文本编辑器。但不选用 MATLAB 的内置编辑/调试器时，不具有调试功能。

【Help Directory】框用于指定帮助信息所在的子目录。在 MATLAB 完成安装之后，用户可将帮助子目录挪到别的硬盘分区上。只要使用该框指定的帮助信息所在目录，就能使用 MATLAB 的联机帮助。

【Echo On】选项可以选择运行 MATLAB 的程序时是否在命令窗口中显示正在运行的命令。选中时显示运行的每条命令。不过此时程序的运行速度变慢。

【Show Toolbar】选项将在 MATLAB 命令窗口的上方显示所有的工具栏按钮，这将极大地方便操作。

【Enable Graphical Debugging】选项决定是否可对图形动态交换进行调试。

②第二张【Command Window Font】选项卡（图 2-21）

该选项卡中的选项控制着命令窗口中字体的类型、大小、风格以及命令窗口的背景色和字体颜色。在【Sample】框内将显示出样例。

③第三张【Copying Options】选项卡（图 2-22）

该选项卡的功能是选择复制 MATLAB 所绘制图形的形式。选中【Clipboard Format】框中的【Windows Metafile】单选钮时，复制为 Windows 图元文件格式；当选中【Windows Bitmap】单选按钮时，则复制为 Windows 位图文件格式。

选中【Honor figure size properties】复选框时，MATLAB 会利用其图形的【Paper Position】属性，根据需要复制纸张的大小，自动调整图形的大小；否则，将以图形的实际大小复制。选中【White background】复选框时，复制图形时将选用白色背景。

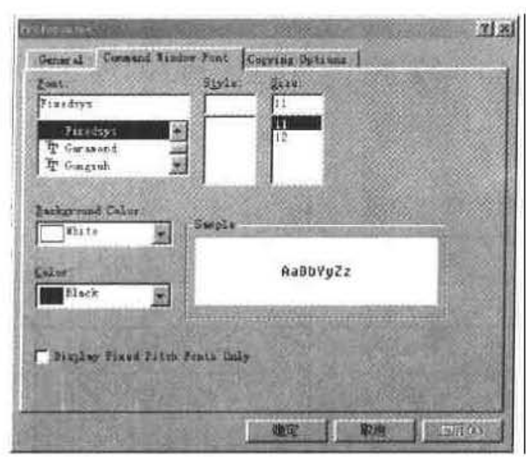


图 2-21 “Command Window Font】选项卡

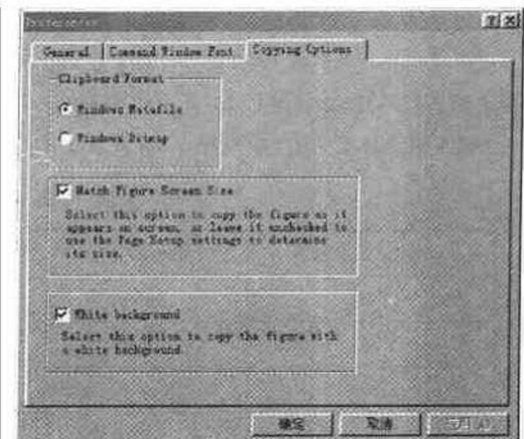


图 2-22 【Copying Options】选项卡

（12）【Print Setup】命令

该项命令用于设置打印机的参数，如打印机的类型、名称、纸张大小、打印方向、图形的打印质量等。

（13）【Print】命令

该项命令用于打印 MATLAB 命令窗口中的内容，也可以设置一些打印参数。

（14）【Print Selection】命令

当选中命令窗口编辑区中的一部分内容后，执行该命令即可按照【Print Setup】选项的设置进行打印。

（15）【Exit MATLAB】命令

单击【Exit MATLAB】命令，将退出 MATLAB。退出 MATLAB 的快捷键为 Ctrl+Q。退出 MATLAB 还可以在其命令窗口中使用命令【quit】或【exit】，均可达到退出的目的。

2. Edit（编辑）菜单

【Edit】菜单项下拉后有 7 项菜单命令，从上到下依次为：【Undo】、【Cut】、【Copy】、【Paste】、【Clear】、【Select All】、【Clear Session】。

【Undo】命令的作用为撤销上一次操作；当选中某一块文档时，可使用【Cut】命令将其剪切到粘贴板上；使用【Copy】命令可以复制选中的文档；使用【Paste】命令可以将粘贴板上的内容复制到指定的位置。这四项命令的功能和其他应用程序中的相应菜单命令没什么区别。

在命令窗口编辑区输入命令出现错误或某些内容不需要时，选中想删除的内容，可以使用【Clear】命令将其清除（已经敲回车键的命令不能清除）。【Select All】命令用于选择命令窗口中的所有内容；当命令窗口中的内容不需要时，可用【Clear Session】命令清除窗口中显示的全部内容（但并不删除工作空间的变量）。

3. View（查看）菜单

【View】菜单的功能是用于控制是否显示工具栏。当选中【Toolbar】项时，工具栏将显示在命令窗口的上方；当不选中【Toolbar】项时，工具栏将被隐藏。

4. Window（窗口）菜单

【Window】窗口菜单命令，可以用来查看目前 MATLAB 打开的所有窗口，并可选中某个窗口以及实现不同窗口之间的切换。

5. Help（帮助）菜单

【Help】帮助菜单命令，下拉有 7 个子菜单项。【Help Window】命令用来打开 MATLAB 的帮助窗口（图 2-23）。可以用鼠标点击窗口中的帮助主题，获得相应问题的详细帮助。【Help Tips】命令也用来打开帮助窗口，并且显示 MATLAB 帮助系统的分类和使用方法。【Help Desk（HTML）】命令用来打开系统中 Microsoft Internet Explorer 浏览器，并显示 MATLAB 的帮助平台（图 2-24）。通过帮助平台可以得到 MATLAB 的详细帮助。

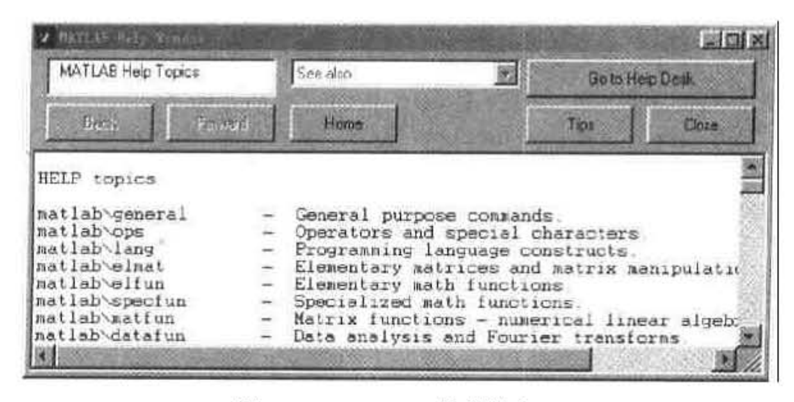


图 2-23　MATLAB 的帮助窗口

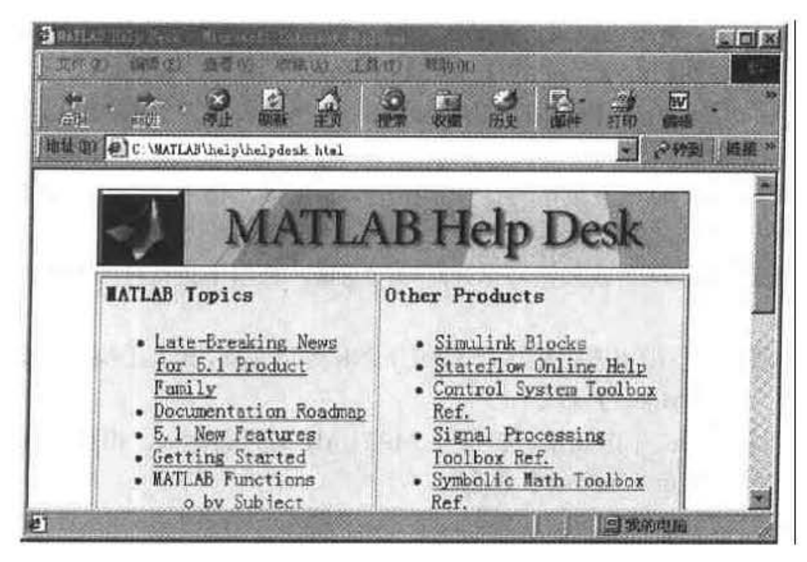


图 2-24　Microsoft Internet Explorer 浏览器中的帮助平台

【Examples and Demos】命令用于演示 MATLAB 提供的示例，来帮助熟悉相关部分的用法。单击【About MATLAB】命令，将打开关于 MATLAB 的版本和版权等信息。单击【Show License】命令将显示 MATLAB 的许可证书。点击【Join MATLAB Access】命

令，将打开 subscribe.html 文件。该文件供联网用户通过先填写其中相关内容的表格，尔后即可连接到 Internet 网上，向 Mathworks 公司申请注册并加入 MATLAB 用户组织。加入用户组织后，可以获得 MATLAB 的各种信息以及非用户组成员无法得到的益处。

2.2.1.2 命令窗口的工具栏

命令窗口中主菜单下面是工具栏（图 2-25），工具栏有 10 个按钮，是 MATLAB 为用户提供的部分常用菜单项命令的快捷方式。熟练使用工具栏按钮，可使计算机操作更快捷高效、更简单方便。



图 2-25 MATLAB 工具栏

图 2-25 所示的工具栏自左到右共有 10 个按钮，其功能如下：

第 1 个【 】：单击该按钮将用 MATLAB 的 M 文件编辑/调试器打开一个新文件，相当于执行在菜单项【File】选中的【New】命令；

第 2 个【 】：单击该按钮将用 MATLAB 的 M 文件编辑/调试器打开一个已存在的 M 文件，相当于执行在菜单项【File】选中的【Open】命令；

第 3 个【 】：单击该按钮将选中的文档剪切到粘贴板上，相当于执行在菜单项【File】选中的【Cut】命令；

第 4 个【 】：单击该按钮将粘贴板上的内容复制到指定位置，相当于执行在菜单项【File】选中的【Copy】命令；

第 5 个【 】：单击该按钮将粘贴板上的文本内容粘贴到命令窗口中的指定位置，相当于执行在菜单项【File】选中的【Paste】命令；

第 6 个【 】：单击该按钮将撤销最近的一次操作，相当于执行在菜单项【File】选中的【Undo】命令；

第 7 个【 】：单击该按钮将打开工作空间浏览器，相当于执行在菜单项【File】选中的【Show Workspace】命令；

第 8 个【 】：单击该按钮将打开路径浏览器，相当于执行在菜单项【File】选中的【Set Path】命令；

第 9 个【 】：单击该按钮将打开 SIMULINK 库浏览器，相当于执行在菜单项【File】选中【New】下的【Model】命令；

第 10 个【 】：单击该按钮将打开 MATLAB 的帮助窗口，相当于执行在菜单项【Help】选中的【Help Windows】命令。

2.2.1.3 命令窗口的命令编辑区

在 MATLAB 的命令窗口中，除标题栏、菜单栏、工具栏以外的窗口，用于输入和显示计算结果，称为命令编辑区。

在 MATLAB 窗口命令编辑区里，可以输入、编辑并运行命令、函数与程序，还可以进行各种不同类型的数学运算。

在 MATLAB 里，有很多的控制键和方向键可用于语句行的编辑。如果能熟练使用这些键将大大提高工作效率。表 2-2 介绍了 MATLAB 的控制键及其作用。

表 2-2　命令窗口的快捷键

键	相应快捷键	功　能
↑	Ctrl+P	光标回调上一行
↓	Ctrl+N	光标回调下一行
←	Ctrl+B	光标同移一个字符
→	Ctrl+F	光标向右移动一个字符
Ctrl →	Ctrl+R	光标向右移动一个单词
Ctrl ←	Ctrl+L	光标向左移动一个单词
Backspace	Ctrl+H	删除光标前字符
Delete	Ctrl+D	删除光标后字符
End 键	Ctrl+E	光标移到行尾
Esc 键	Ctrl+U	删除一行
Home 键	Ctrl+A	光标移到行首
	Ctrl+K	从光标处一直删除到行尾
注：Ctrl+E 表示同时按下 Ctrl 键和 E 键，其余类推。		

在 MATLAB 的命令窗口里，在语句行尾加上 3 个英文句号表示续行；运算符“=”、“+”、“-”前后的空格不影响计算结果；在输入命令后以分号结束，则不会显示命令执行结果，但可使程序运行速度大大提高，这对有大量输出数据的程序特别有用。还可用键 Ctrl+C 来终止正在运行的程序。

在 MATLAB 的命令窗口里，对于选中的语句行、程序或部分程序块等，可用复制、剪切与粘贴等菜单项命令进行操作，用鼠标右键将其移动等。用这些手段可以对命令窗中的内容进行各种不同的编辑。

2.2.2 MATLAB 的工作空间

所谓工作空间（Workspace）实际上是运行 MATLAB 的程序或命令时，所保存的所有变量与 MATLAB 提供的常量所构成的空间。每打开一次 MATLAB 时，系统就自动打开工作空间，在 MATLAB 运行期间工作空间一直都是打开的，关闭 MATLAB 时该工作空间也同时关闭。刚打开的 MATLAB 的工作空间中只有 MATLAB 提供的几个常量，如 pi（3.1415926...）、虚数单位 i 等。当运行 MATLAB 程序时，程序中的变量被存入到工作空间中。除非用特殊的命令删除某变量，否则该变量在关闭 MATLAB 之前一直存在。

一个程序中的运算结果以变量的形式保存在工作空间中，工作空间还可被别的程序继续利用；某个时刻工作空间中的所有变量还可以存储到一个文件中，以供不时之需。

可以随时查看工作空间中的变量名及变量的值。在命令窗口键入 who 和 whos 命令可以看到目前工作空间里的所有变量。Who 命令只给出变量名；whos 还给出变量的大小、数据类型等内容。命令 clear 可删除工作空间里的变量。

2.2.3 MATLAB 的变量浏览器

工作空间（ Workspace）是存放变量有关信息的，所以变量浏览器就是工作空间浏

览器。以上已经说明，执行在 MATLAB 主菜单的【File】下的【Show Workspace】子菜单项命令，或者利用命令窗口工具栏的【Show Workspace】按钮，再或者在命令窗口中运行 workspace 命令，均可打开变量浏览器（图 2-26）。

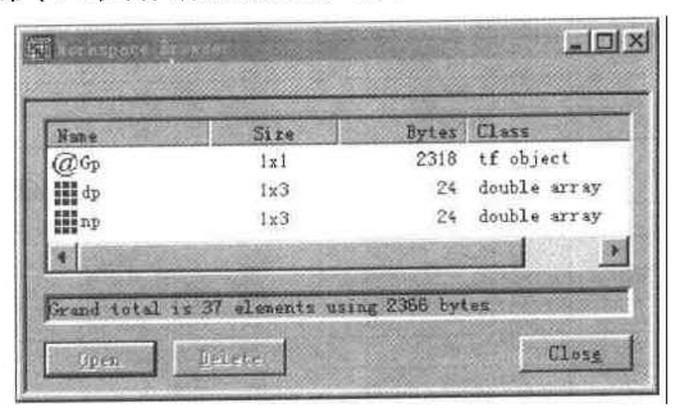


图 2-26　MATLAB 变量浏览器

由图 2-26 可见，变量浏览器用于显示工作空间里变量的属性：变量名、变量数组维数、占用字节数、变量类别（单精度或双精度）、是否复数等。另外，在变量浏览器里可以用【Delete】按钮删除被选中的变量；也可以用【Open】按钮打开变量编辑器（图 2-27），还可以在变量浏览器里对选中的变量用鼠标左键双击来打开变量编辑器，查看变量的属性或对变量元素进行修改。

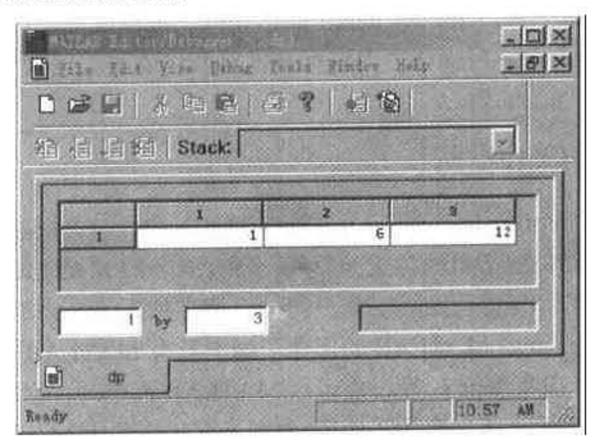


图 2-27　MATLAB 变量编辑器

2.2.4　MATLAB 的程序编辑器

MATLAB 提供了一个程序编辑/调试器（图 2-28）。在 MATLAB 的命令窗口中有三种方式可进入程序编辑/调试器：

（1）择菜单栏的【File】项中的【New】或【Open】项。

（2）选择工具栏的【New】或【Open】按钮。

（3）在命令编辑区中输入 edit 命令。

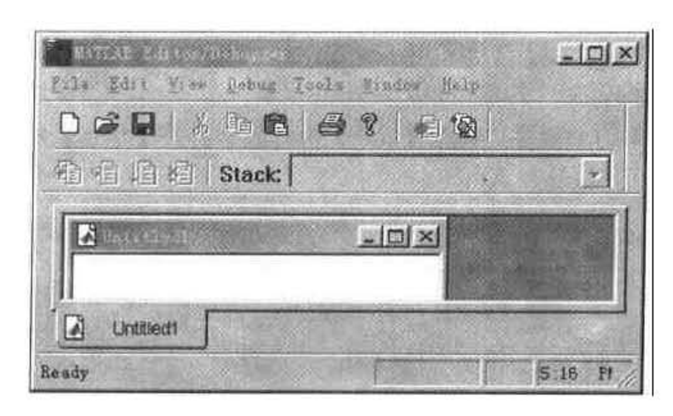


图 2-28　程序编辑/调试器

已经介绍过，在【General】选项卡中，若选择【Editor Preference】为【Built-in Editor】时，则表明选择 MATLAB 的内置程序编辑/调试器。

在程序编辑/调试器里，不同的文本内容分别用不同颜色的字体显示。其中 MATLAB 的关键字如 for、if、end 等为蓝色，注释语句为绿色，字符串为红色，其他文本为黑色，这样的颜色设置非常醒目，便于调试。

在程序编辑/调试器的【View】菜单项里有两个常用的命令：【Evaluate Selection】和【Auto Indent Selection】，对于编辑或调试程序非常有用。当选定一部分程序后，再选择【Evaluate Selection】项，MATLAB 会计算所选部分程序的值并在命令窗口里显示结果。当选定程序的一部分后再选择【Auto Indent Selection】项，程序编辑/调试器就会根据程序的逻辑关系自动编排格式，这样程序看起来就更清楚明了。

2.2.5　MATLAB 的路径浏览器

路径浏览器能够修改 MATLAB 的搜索路径并查看任一路径下的所有文件。点击命令窗口工具栏的路径浏览器按钮（【】）或选择【File】菜单条下的【Set Path】项，均可打开路径浏览器（图 2-19）。

路径浏览器主菜单【File】的下拉菜单命令【Open】、【Save Path】、【Exit Path Browser】，分别用来在路径浏览器中打开 M 文件、保存路径及退出路径浏览器。

路径浏览器主菜单【Edit】的下拉菜单项命令【Undo】、【Redo】分别用来在路径浏览器中进行撤销与恢复操作。

路径浏览器主菜单【View】的下拉菜单命令【Toolbar】、【Status Bar】、【Editor/Debugger】、【Path Browser】、【Workspace Browser】分别用来在路径浏览器中设置或隐藏工具栏、设置或隐藏状态栏，并可以显示 MATLAB 的编辑/调试器、路径浏览器与变量浏览器。

路径浏览器主菜单【Path】命令用来执行对搜索路径的操作。

路径浏览器主菜单【Tools】为用户提供了一些有用的工具。下拉菜单命令【Run】用于直接运行被选中的文件。下拉菜单命令【Customize】执行后会打开其对话框，用这个对话框用户能够自定义工具菜单。

路径浏览器主菜单【Tools】的下拉菜单命令【Options】执行后会打开其对话框，其

中有两张选项卡，【General】选项卡用于设置路径浏览器的常规属性；【Editor】选项卡用于编辑 M 文件时设置文件的某些相关属性。

主菜单【Tools】的下拉菜单命令【Font】执行后会打开【字体】对话框（图 2-29），用来设置字体的类别、字体样式与字体大小。

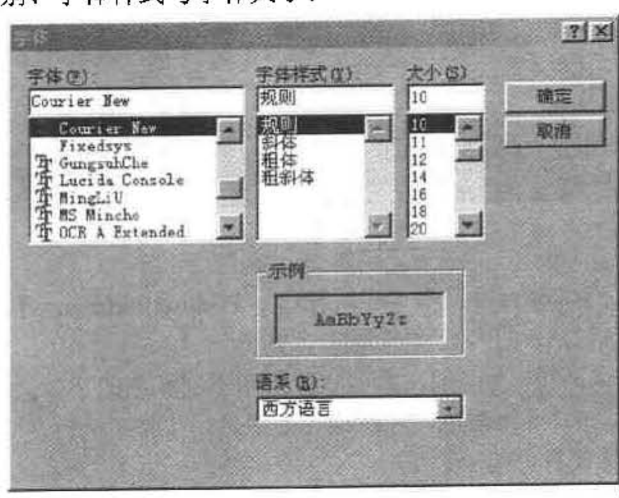


图 2-29 【字体】对话框

路径浏览器的【Current" 文本框，用户可以在其中改变 MATLAB 的当前目录，也可以再用旁边的【Browse】按钮选择要查询的目录，当然目录是在某一路径下的。"Path"文本框内列出了 MATLAB 搜索路径中的所有目录；而"Files in…"文本框内，既可以查看"Current"文本框内所确定目录下的文件，也可以查看"Path"文本框内所确定路径下的文件，这需要用鼠标点击"Current"或"Path"文本框内带路径的目录。

"Current"文本框内所显示的文件，选中后双击之，就可以打开该文件。例如在路径 MATLABR11\toolbox\control 下，可以打开 MATLAB 中的"bode.m"M 文件（图 2-30）。

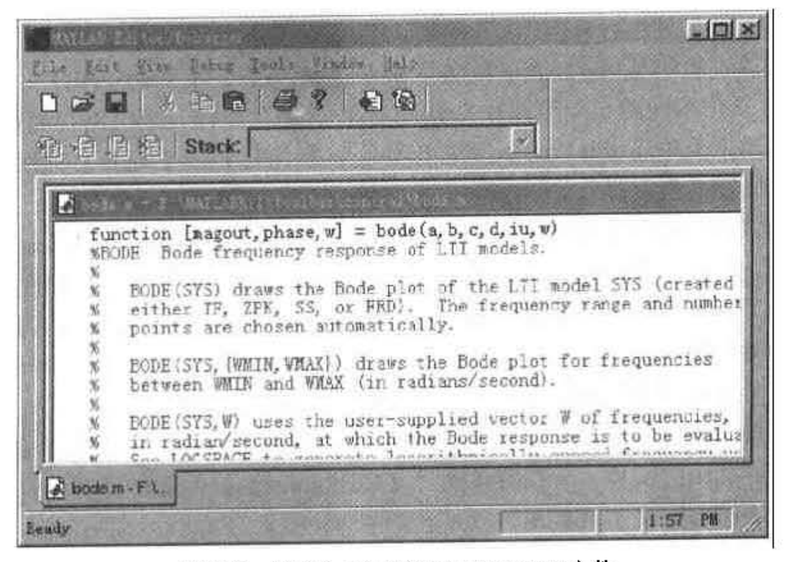


图 2-30 MATLAB 中的"bode.m"M 文件

2.2.6 MATLAB的Work子目录

MATLAB系统的目录结构中有个子目录“Work”，它是MATLAB默认存储文件的地方。当在MATLAB命令窗口中存储或者打开文件时，只要没有设置文件路径与子目录，MATLAB系统就默认为是“MATLABR11\Work”。

所以，MATLAB系统下的子目录“Work”，是一个既有用又方便的“舞台”，将设计或欲调试的M文件、MATLAB程序存储在这里是最佳选择。在MATLAB命令窗口中存储或者打开文件时，可以省略路径，这会给编辑、输入、调试程序以及仿真带来很大的方便。

2.2.7 MATLAB运行外部程序

如果要运行外部程序例如要进行DOS操作或是利用DOS操作系统的编辑器编辑文件，按常规作法必须先退出MATLAB系统，尔后进入DOS系统才能操作。MATLAB提供了一个特殊命令“！”，允许不退出本身的运行就可以进入DOS窗口而进行DOS操作。这只要在MATLAB命令窗口中键入感叹号“！”，回车后，即会打开一个DOS窗口（参见图2-31）。

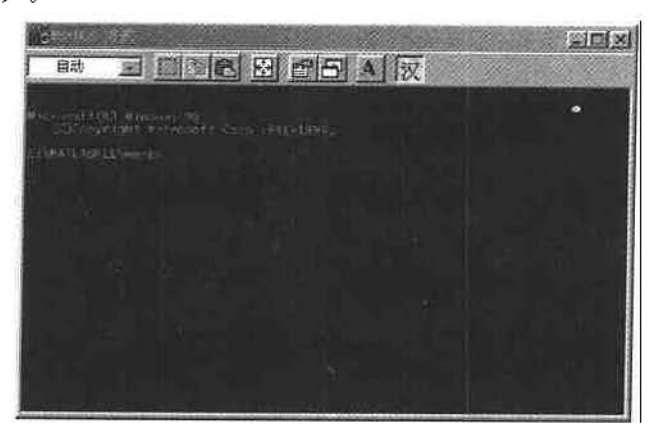


图2-31　命令窗口中运行“!”产生的DOS窗口

可以在图2-31所示的DOS窗口中进行DOS操作。还可以利用DOS操作系统的编辑器（EDIT）对MATLAB的M文件进行编辑与修改。图2-32为利用DOS操作系统的编辑器正在编辑的函数cqjz.( )。

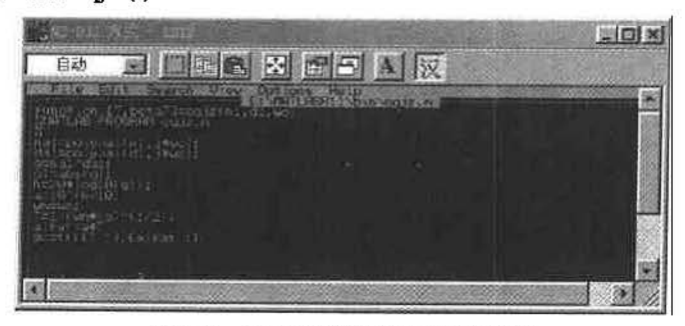


图2-32　DOS系统编辑器里编辑程序

2.3 MATLAB 数值运算基础

MATLAB 数值运算的表达式变量中不允许有未定义的自由变量，MATLAB 的数值运算主要是指数组的运算与矩阵的运算。不需要事先定义，MATLAB 系统定义的数值元素是复数，这是 MATLAB 区别其他高级语言的最大特点之一，它给诸多领域的数值计算带来极大的方便。

控制系统的 MATLAB 计算与仿真中有大量的数值计算，本节举例是为了熟悉并掌握 MATLAB 运算表达式的格式。

2.3.1 常量与变量

1．常量

MATLAB 中使用的数据有常量与变量。常量有实数常量与复数常量两类，复数由实部与虚部组成，当然可以把实数常量当作复数常量的虚部为零的特例。常量可以使用传统的十进制计数法表示，也可以使用科学计数法来表示。

在 MATLAB 中，虚数单位 i = sqrt(-1)，在工作空间内显示的值为：

```
i =
ans =
    0 + 1.0000i
```

复数常量的生成可以利用如下语句：

$z = a + b\,\mathrm{i}$

或

$z = r * \exp(\theta * \mathrm{i})$，其中 r 是复数的模，θ 是复数幅角的弧度数。

在 MATLAB 中，所有常量数值在内存的存储格式都是使用 IEEE 标准规定的 16 位长型格式，其数值的有效范围是 $10^{-308} \sim 10^{+308}$。

2．变量

MATLAB 里的变量无须事先定义。一个程序中的变量，以其名称在语句命令中第一次合法出现而定义。请注意 MATLAB 变量名称的命名不是任意的，其命名规则如下：

（1）变量名可以由英语字母、数字和下划线组成。

（2）变量名应以英语字母开头。

（3）组成变量名的字符长度不大于 31 个。

（4）MATLAB 区分大小写英语字母。

MATLAB 中设置了一些特殊的变量与常量，现列于表 2-3 中。

表 2-3 MATLAB 的特殊变量与常量

变 量 名	功 能 说 明
ANS（或 ans）	缺省变量名，以应答最近一次操作运算结果
i 或 j	虚数单位，定义为 $\sqrt{-1}$
pi	圆周率

（续）

变量名	功能说明
eps	浮点数的相对误差
realman	最大的正实数
realmix	最小的正实数
INF（或 inf）	代表无穷大
NaN（或 nan）	代表不定值（即 0/0）
nargin	函数实际输入参数个数
nargout	函数实际输出参数个数

MATLAB 中还可以设置一种特殊的变量——全局变量。这是因为在 MATLAB 里进行编程的过程中，有时会需要某个变量既作用在主程序里又作用在调用的子程序里，或者需要某个变量作用在多个函数中，这时就可以将该变量设置成全局变量。

只要在该变量前添加 MATLAB 的关键字"global"就可将该变量设定为全局变量了。全局变量必须在使用前声明，即这个声明必须放在主程序的首行；而且作为一个惯用的规则，在 MATLAB 程序中应尽量用大写字母书写全局变量。

2.3.2 MATLAB 运算符

1. MATLAB 算术运算符

MATLAB 算术运算符如表 2-4 所示。

表 2-4 MATLAB 算术运算符

操作符	功能说明	操作符	功能说明
+	加	\	矩阵左除
-	减	.\	数组左除
*	矩阵乘	/	矩阵右除
.*	数组乘	./	数组右除
^	矩阵乘方	'	矩阵转置
.^	数组乘方	.'	数组转置

2. MATLAB 关系运算符

MATLAB 关系运算符如表 2-5 所示。

3. MATLAB 逻辑运算符

MATLAB 的逻辑运算符如表 2-6 所示。

表 2-5 MATLAB 关系运算符

操作符	功能说明
==	等于
~=	不等于
>	大于
<	小于
>=	大于等于
<=	小于等于

表 2-6 MATLAB 逻辑关系运算符

逻辑运算符	逻辑运算	说明
&	And	逻辑和
|	Or	逻辑或
~	Not	逻辑非
	Xor	逻辑异或

4. MATLAB 特殊运算符

MATLAB 特殊运算符如表 2-7 所示。需要特别指出，下表中的特殊运算符是指在英语输入状态下输入的字符，若在汉字输入状态下输入则无效。

表 2-7 MATLAB 特殊运算

符号	功能说明
:	冒号，在 MATLAB 中非常重要
;	分号，用于分隔行
,	逗号，用于分隔列
( )	圆括号，指出在算术表达式中先后次序
[ ]	方括号，用于构成向量和矩阵
{ }	大括号，用于构成单元数组
.	小数点或域访问符
..	父目录
...	用于语句行尾端表示该行未完
%	用于注释
!	用于调用操作系统命令
=	用于赋值

经常使用的 MATLAB 特殊运算符，功能说明如下。

（1） 冒号（：）

①使用冒号可以输入行向量，其格式为：$x = a : inc : b$。其中，a 是数组的第一个元素，inc 是步长（两个元素的间隔），步长可正可负；若（b-a）为 inc 的整数倍时，数组的最后一个元素等于 b，否则小于 b；a、inc、b 三者之间用冒号分隔；当 inc=1 时可以省略。

【例 2-1】冒号使用示例。

在 MATLAB 命令窗口输入指令并回车（以下举例同此）。

```
a=10:5:30
```

当指令回车后，显示结果如下：

```
a =
    10    15    20    25    30
a=10:-1:0
```

当指令回车后，显示结果如下：

```
a =
    10     9     8     7     6     5     4     3     2     1     0
a=10:15
```

当指令回车后，显示结果如下：

```
a =
    10    11    12    13    14    15
```

②使用冒号能够从向量、矩阵和数组中选出指定元素、行和列。

【例 2-2】使用冒号选出指定元素。

```
A=[1 2 3;4 5 6;7 8 9]
A (1:2,3)
```

指令回车后，结果为从矩阵 *A* 中取第一行和第二行的第三列元素构成的列向量：

```
ans =
     3
     6
```

③使用冒号能够从一个大矩阵中选取小矩阵。

【例 2-3】使用冒号选取小矩阵。

```
A=[1 2 3;4 5 6;7 8 9];
B=A (2:3,:)
```

回车后，结果为由矩阵 *A* 的第二、三行组成的矩阵 ***B***：

```
B =
     4     5     6
     7     8     9
```

（2） 方括号（[]）

方括号可以输入矩阵，也可以用方括号删除矩阵的行和列,此时方括号[]代表空阵。

【例 2-4】用方括号删除矩阵的行和列。

```
A=magic (4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
A (:,3) =[ ];
A
```

指令回车后，得到结果为：

```
A =
    16     2    13
     5    11     8
     9     7    12
     4    14     1
```

可以发现，这个 4 阶魔术方阵的第 3 列元素被删除了。

2.3.3 矩阵、数组及其算术运算

MATLAB 是“Matrix Laboratory”之意，即矩阵实验室。MATLAB 最初是为解算线性代数的问题而开发的，MATLAB 以矩阵作为基本的运算单元。矩阵是在线性代数中定义的。什么是矩阵？

线性代数中矩阵是这样定义的：有 $m \times n$ 个数 $a_{ij}(i=1,2,\cdots,m; j=1,2,\cdots n)$ 的数组将其排成如下格式（用方括号括起来）的“表”：

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

此表作为整体，将它当作一个抽象的量称为矩阵，且是 m 行 n 列的矩阵。横向每一行所有元素依次序排列则为行向量；纵向每一列所有元素依次序排列则为列向量。请特别注意，数组用方括号括起来后已作为一个抽象的特殊量——矩阵。在线性代数中，矩阵有特定的数学函义，并且有其自身严格的运算规则。矩阵概念是线性代数范畴内特有的。

在 MATLAB 中，定义了矩阵规则及其运算符。MATLAB 中的矩阵运算规则与线性代数中的矩阵运算规则相同。

数组（Array）是由一组复数排成的长方形阵列（而实数可视为复数的虚部为零的特例）。对于发展了的 MATLAB，在线性代数范畴之外，数组也是进行数值计算的基本处理单元。一行多列的数组是行向量；一列多行的数组就是列向量；数组可以是二维的“矩形”，也可以是三维的，甚至还可以是多维的。多行多列的“矩形”数组与数学中的矩阵从外观形式与数据结构上看，没有什么区别。

在 MATLAB 中，也定义了一套数组运算规则及其运算符，但数组运算是 MATLAB 软件所定义的规则，定义规则是为了管理数据方便、操作简单、指令形式自然、程序简单易读与运算高效。在 MATLAB 中的大量数值计算是以数组形式进行的。而在 MATLAB 中凡是涉及线性代数范畴的问题，其运算则是以矩阵作为基本的运算单元。

MATLAB 既支持数组的运算也支持矩阵的运算。但在 MATLAB 中，数组与矩阵的运算却有很大的差别。在 MATLAB 中，数组的所有运算都是对被运算数组中的每个元素平等地执行同样的操作。矩阵运算是从把矩阵整体当作一个特殊的量这个基点出发，依照线性代数的规则来进行的运算。

2.3.3.1 数组或矩阵元素的标识

为在 MATLAB 中对数组与矩阵熟练地进行运算，必须对数组与矩阵元素的定位即元素的标识非常熟悉。

1．一维数组元素的标识、访问与赋值

一维数组（行向量）是使用方括号以及在括号内列出以空格或逗号分隔其元素的表。一维数组的元素是以数组名后圆括号内的元素在数组中位置的顺序号来标识的，数组元素的访问与赋值就是根据数组元素的标识进行的。

【例 2-5】一维数组元素的标识。

在 MATLAB 命令窗口输入以下指令并回车（以下举例同此）。

```
X=[1*pi 2*pi 3*pi 4*pi 5*pi]
X =
    3.1416    6.2832    9.4248   12.5664   15.7080
```

如查询 X 数组的第 3 个元素可输入以下指令并回车：

```
X(3)
ans =
```

```
   9.4248
```

查询 X 数组的第 2~4 个元素：

```
X(2:4)
ans =
   6.2832    9.4248   12.5664
```

查询 X 数组的第 4 到最后一个元素：

```
X(4:end)
ans =
  12.5664   15.7080
```

查询 X 数组的第 3、2、1 个元素：

```
X(3:-1:1)
ans =
   9.4248    6.2832    3.1416
```

查询 X 数组中＜10 的元素：

```
X(find(X<10))
ans =
   3.1416    6.2832    9.4248
```

查询 X 数组的第 4、2、5 个元素：

```
X([4 2 5])
ans =
  12.5664    6.2832   15.7080
```

将 X 数组的第 1 个元素重新赋值为 1：

```
X(1)=1;
X
X =
   1.0000    6.2832    9.4248   12.5664   15.7080
```

2．多维数组或矩阵元素的标识、访问与赋值

由于多行多列的“矩形”数组跟矩阵的外观形式与数据结构相同，所以多维数组元素的标识即多维数组元素定位地址就是矩阵元素的标识或定位地址。其元素标识的通用双下标格式如下：

A（m,n）

其中，*m* 为行号；*n* 为列号。有了元素的标识方法，对于多维数组（或矩阵）元素的访问与赋值常用的相关指令格式如表 2-8 所列。

表 2-8　子数组访问与赋值常用的相关指令格式

指令格式	指 令 功 能
A（*r*,*c*）	由数组 *A* 中指定行 *r*、指定列 *c* 之元素组成的子数组
A（*r*,:）	由数组 *A* 中指定行 *r* 对应的所有列之元素组成的子数组
A（:,*c*）	由数组 *A* 中指定列 *c* 对应的所有行之元素组成的子数组

（续）

指令格式	指令功能
A（:）	由数组 A 的各个列按从左到右的次序首末相接的“一维长列”子数组
A（i）	“一维长列”子数组的第 i 个元素
A（r,c）=Sa	对数组 A 赋值，Sa 也必须为 Sa（r,c）
A（:）=D（:）	数组全元素赋值，保持 A 的行宽、列长不变，A、D 两数组元素总数应相同，但行宽、列长可不同

【例 2-6】数组（或矩阵）元素的标识。

查询 A 数组第 2 行、第 3 列的元素

```
A=[1 2 3;4 5 6;7 8 9];
A(2,3)
ans =
     6
```

查询 A 数组第 3 行所有的元素

```
A(3,:)
ans =
     7     8     9
```

查询 A 数组第 2 列转置后所有的元素

```
(A(:,2))'
ans =
     2     5     8
```

查询 A 数组按列拉长转置后所有的元素

```
(A(:))'
ans =
     1     4     7     2     5     8     3     6     9
```

查询“一维长列”A 数组第 6 个元素

```
A(6)
ans =
     8
```

查询原 A 数组所有的元素

```
A
A =
     1     2     3
     4     5     6
     7     8     9
```

创建 S 数组所有的元素

```
S=[5 5 5;5 5 5;5 5 5]
S =
     5     5     5
     5     5     5
```

```
    5     5     5
```

以“双下标”方式对数组 *A* 赋值

```
A=S
A =
    5     5     5
    5     5     5
    5     5     5
```

创建 *T* 数组所有的元素

```
T=[8 8 8 8 8 8 8 8 8]
T =
    8     8     8     8     8     8     8     8     8
```

以数组全元素赋值方式对数组 *A* 赋值

```
A(:)=T(:)
A =
    8     8     8
    8     8     8
    8     8     8
```

2.3.3.2 数组与矩阵的输入

一行多列的数组是行向量，矩阵横向行的所有元素依次序排列的也是行向量。以下介绍行向量的输入法。

1．一维行或列向量的输入

（1）显示元素列表输入

【例 2-7】向量元素的列表输入。

在 **MATLAB** 命令窗口输入指令并回车（以下举例同此）。

```
A=[1 2*pi sqrt(3) 4+5i]
```

指令回车后得到结果

```
A =
  1.0000 6.2832 1.7321 4.0000 + 5.0000i
B=[1 2 3]'
```

指令回车后得到结果

```
B =
     1
     2
     3
```

（2）冒号生成法

一般格式为：

x=a:inc:b

有关格式说明参见冒号运算符功能。

【例 2-8】冒号生成向量。

t=0:0.1:0.6 与 t=[0:0.1:0.6]两指令运行结果一样：

```
t =
    0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
```

（3）常量线性分隔生成法

一般格式为：

x=linspace(a,b,n)

格式说明：*a*,*b* 分别是生成数组的第一个与最后一个元素；*n* 是分隔的总间隔数。这个 MATLAB 函数与指令 *x=a*:（*b-a*）/（*n-1*）: *b* 等效。

【例 2-9】线性分隔生成向量。

x=linspace(0,0.6,6) 与 x=0:(0.6/5):0.6 两指令运行结果一样：

```
x =
    0    0.1200    0.2400    0.3600    0.4800    0.6000
```

（4）常量对数分隔生成法

一般格式为：

x=logspace(a,b,n)

格式说明：生成数组的第一个元素为10^a，最后一个元素为10^b。*n* 是分隔的总间隔数。

【例 2-10】对数分隔生成向量。

w=logspace(0,2,10)

指令其运行结果为：

```
w =
  Columns 1 through 7
    1.0000    1.6681    2.7826    4.6416    7.7426   12.9155   21.5443
  Columns 8 through 10
   35.9381   59.9484  100.0000
```

2. 二维数组（或矩阵）的输入

（1）显示元素列表输入

在 MATLAB 中输入数组需要遵循以下基本规则：

①把数组元素列入括号[ ]中。

②每行内的元素间用逗号或空格分开。

③行与行之间用分号或回车键【Enter】隔开。

【例 2-11】元素列表输入数组（或矩阵）。

在 MATLAB 命令窗口输入指令并回车（以下举例同此）。

```
A=[1 2 3;4 5 6;7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
```

（2）利用 M 文件生成数组或矩阵

【例 2-12】利用 M 文件生成矩阵

如果在 MATLAB 文件编辑/调试器里有一 M 文件 L2301.m，内容如下：

```
% MATLAB FILE L2301.m
E=[11 12 13;21 22 23;31 32 33]
```

在 MATLAB 中运行 L2301.m 文件后有矩阵 ***E***：

```
E =
    11    12    13
    21    22    23
    31    32    33
```

（3）小矩阵联接生成大矩阵

在 MATLAB 中，利用联接算子——方括号[]可将小矩阵联接为一个大矩阵。

【例 2-13】利用方括号[]将小矩阵联接成大矩阵举例。

```
A=[1 2;3 4];
A1=A+5
```

当指令运行后可生成矩阵：

```
A1 =
     6     7
     8     9
A2=A+10
```

当指令运行后可生成矩阵：

```
A2 =
    11    12
    13    14
A3 =A1+10
```

当指令运行后可生成矩阵：

```
A3 =
    16    17
    18    19
G=[A A1;A2 A3]
```

当指令运行后可生成矩阵：

```
G =
     1     2     6     7
     3     4     8     9
    11    12    16    17
    13    14    18    19
```

由上可见，4 个 2×2 的子矩阵组成一个 4×4 的矩阵 ***G***。

2.3.3.3 数组、矩阵的算术运算

前面已经说明，数组无论作什么运算，总是对被运算数组中的每一个元素进行同等

的操作。矩阵运算则不相同，它是把矩阵当作一个整体，依照线性代数的规则进行运算。

1．数组、矩阵的加减运算

数组加减运算和矩阵加减运算的条件都是两个数组或矩阵的行数与列数分别相同，其运算规则也是相同的，即都是数组相应元素或矩阵相应元素的加减运算。

在 MATLAB 里，维数为 1×1 的数组叫做标量。而 MATLAB 里的数值元素是复数，所以一个标量就是有一个复数。

需要指出的是，标量与数组间可以进行加减运算，其规则是标量与数组的每一个元素进行加减操作。矩阵与标量间不存在这种运算。

【例 2-14】数组与标量之间实施加减运算。

在 MATLAB 命令窗口输入指令并回车（以下举例同此）。

```
s=5;
A=[1 2 3;4 5 6;7 8 9];
A+s
```

指令运行结果为：

```
ans =
     6     7     8
     9    10    11
    12    13    14
A-s
```

指令运行结果为：

```
ans =
    -4    -3    -2
    -1     0     1
     2     3     4
s-A
```

指令运行结果为：

```
ans =
     4     3     2
     1     0    -1
    -2    -3    -4
```

2．数组、矩阵的乘法运算

数组的乘法是用运算符“.*”表示，即在乘号前加一个点来特别指定是数组的乘法运算。数组的乘法运算必须在具有相同维数的数组间进行，其结果是数组的对应元素间相乘的结果组成的新数组。而两矩阵相乘必须服从数学中矩阵叉乘的条件与规则。

（1）数组、矩阵与标量的乘法运算

数组与一个标量之间或矩阵与一个标量之间的乘法运算都是指该数组或矩阵的每个元素与这个标量分别进行乘法运算。

【例 2-15】标量与数组（或矩阵）之间的乘法运算。

```
s=5;
```

```
A=[1 2 3;4 5 6;7 8 9];
```

s.*A、A.*s 与 s*A、A*s 运算结果都是一样的：

```
ans =
     5    10    15
    20    25    30
    35    40    45
```

（2）数组、矩阵的乘法运算

数组的乘法运算必须在具有相同维数的数组间进行，两矩阵相乘的条件是左矩阵的列数必须等于右矩阵的行数，矩阵乘法不满足交换律。

【例 2-16】数组乘法运算。

```
a=[1 2 3;4 5 6;7 8 9];
b=[2 2 2;2 2 2;2 2 2];
a.*b 或 b.*a
```

两指令运算结果均为：

```
ans =
     2     4     6
     8    10    12
    14    16    18
```

【例 2-17】矩阵乘法运算。

```
a=[1 2 3;4 5 6;7 8 9];
b=[2 2 2;2 2 2;2 2 2];
a*b
```

指令运行结果为：

```
ans =
    12    12    12
    30    30    30
    48    48    48
    b*a
```

指令运行结果为：

```
ans =
    24    30    36
    24    30    36
    24    30    36
```

由此可见，在 MATLAB 中数组的乘法与矩阵的乘法遵循的运算规则是不同的，其结果也就不同。数组乘法 a.*b 与 b.*a 的结果相同；矩阵乘法 a*b 与 b*a 两者运算结果不同，这两个矩阵乘法与数组乘法运算又不同。

3．数组、矩阵的除法运算

（1）数组与标量间的除法运算

也要指出，标量与数组间可以进行除法运算，其规则是标量与数组的每一个元素进

行除法操作。矩阵与标量间则无这种运算。

【例 2-18】数组与标量之间的除法运算。

```
s=5;
A=[1 2 3;4 5 6;7 8 9];
s./A
```

指令运行结果为：

```
ans =
    5.0000    2.5000    1.6667
    1.2500    1.0000    0.8333
    0.7143    0.6250    0.5556
A.\s
```

指令运行结果为：

```
ans =
    5.0000    2.5000    1.6667
    1.2500    1.0000    0.8333
    0.7143    0.6250    0.5556
A./s
```

指令运行结果为：

```
ans =
    0.2000    0.4000    0.6000
    0.8000    1.0000    1.2000
    1.4000    1.6000    1.8000
```

（2）数组、矩阵的除法运算

数组除法与矩阵除法的运算规则也是不相同的。维数相同两数组的除法也是对应元素间的相除，数组的除法没有左除和右除，即符号“.\”和“./”的运算结果是一致的，不过要注意被除数与除数在两种除法运算符中的左右位置是不同的。

矩阵除法运算有左除与右除之分，即运算符号“\”和“/”指代的运算。其运算规则是：$a\backslash b$=inv（a）$*b, a/b=a*$inv（b）。

【例 2-19】数组的除法。

```
a=[1 2 3;4 5 6;7 8 9];
b=[2 2 2;2 2 2;2 2 2];
a./b 或b.\a
```

两指令运行结果均为：

```
ans =
0.5000    1.0000    1.5000
2.0000    2.5000    3.0000
3.5000    4.0000    4.5000
```

【例 2-20】矩阵的除法。

```
c=[1 2 3;4 5 6;7 8 9];
```

```
d=[1 0 0;0 2 0;0 0 3];
c\d
```

指令运行结果为:

```
ans =
  1.0e+016 *
   -0.4504    1.8014   -1.3511
    0.9007   -3.6029    2.7022
   -0.4504    1.8014   -1.3511
inv(c)*d
```

指令运行结果为:

```
ans =
  1.0e+016 *
   -0.4504    1.8014   -1.3511
    0.9007   -3.6029    2.7022
   -0.4504    1.8014   -1.3511
c/d
```

指令运行结果为:

```
ans =
    1.0000    1.0000    1.0000
    4.0000    2.5000    2.0000
    7.0000    4.0000    3.0000
c*inv(d)
```

指令运行结果为:

```
ans =
    1.0000    1.0000    1.0000
    4.0000    2.5000    2.0000
    7.0000    4.0000    3.0000
d\c
```

指令运行结果为:

```
ans =
    1.0000    2.0000    3.0000
    2.0000    2.5000    3.0000
    2.3333    2.6667    3.0000
inv(d)*c
```

指令运行结果为:

```
ans =
    1.0000    2.0000    3.0000
    2.0000    2.5000    3.0000
    2.3333    2.6667    3.0000
```

运算结果说明:第一，矩阵右除、左除是不一样的；第二，矩阵除法运算规则是 c/d=c*inv(d),c\d=inv(c)*d。

4．数组、矩阵的乘方运算

数组的乘方使用运算符“.^”来表示。

（1）数组与标量的乘方运算

①以数组为底而以标量为指数

【例 2-21】以数组为底以标量为指数的乘方运算。

```
a=[4 5 6];
a.^4
```

指令运行结果为：

```
ans =
   256   625   1296
b=[1 2;3 4];
b.^3
```

指令运行结果为：

```
ans =
     1     8
    27    64
```

可以看出，这种运算的规则是以数组中的每个元素为底，分别与作为指数的标量进行乘方运算得到的一个新的数组。

②以标量为底而以数组为指数

【例 2-22】以标量为底而以数组为指数的乘方运算。

```
a=[4 5 6];
b=[1 2;3 4];
d=2;
e=d.^a
f=d.^b
```

指令运行结果为：

```
e =
    16    32    64
f =
     2     4
     8    16
```

可以看出，这种运算的规则是以该标量为底，用数组中的每个元素分别作为指数与该标量进行乘方运算后得到的一个新数组。

（2）数组与数组的乘方运算

【例 2-23】数组与数组的乘方运算示例。

```
a=[4 5 6];
b=[1 2 3];
```

```
c=[2 3;5 6];
d=[1 2;3 4];
e=a.^b
f=d.^c
```

指令运行结果为：

```
e =
     4    25   216
f =
     1           8
   243        4096
```

由此看出，数组的乘方运算规则是以前一个数组为底，后一个数组为指数，其对应的元素分别进行指数运算得到的结果。显然，数组间的乘方运算只在维数相同的数组间进行。

（3）矩阵的乘方运算

【例 2-24】矩阵的乘方运算示例。

```
a=[1 2;3 4];
b=2;
c=[0.5];
d=-0.2;
a^b
```

指令运行结果为：

```
ans =
     7    10
    15    22
```

由此可见，***a*** 为矩阵，*b* 为整数，矩阵的乘方 ***a^b*** 是矩阵 ***a*** 自乘 *b* 次。若 *b*=2,则 a^b=a*a。

```
a^c
```

指令运行结果为：

```
ans =
   0.5537 + 0.4644i   0.8070 - 0.2124i
   1.2104 - 0.3186i   1.7641 + 0.1458i
a^d
```

指令运行结果为：

```
ans =
   0.9210 - 0.5451i  -0.0945 + 0.2494i
  -0.1417 + 0.3740i   0.7793 - 0.1711i
```

由此可见，***a*** 为矩阵，*c* 为标量，矩阵的乘方 ***a^c*** 是矩阵 ***a*** 的 *c* 次方。

除此之外，还可以进行标量的矩阵乘方运算与标量的数组乘方运算。

```
b^a
```

指令运行结果为：

```
ans =
   10.4827   14.1519
   21.2278   31.7106
b.^a
```

指令运行结果为：

```
ans =
     2     4
     8    16
```

5．数组、矩阵的转置运算

对于复数元素，数组的转置运算是非共轭的转置，而矩阵的转置运算则是共轭的转置。

【例 2-25】矩阵与数组的转置运算示例。

```
e=[1 2 3;4 5 6]
e =
     1     2     3
     4     5     6
E=e*(1+i)
```

指令运行结果为：

```
E =
   1.0000 + 1.0000i   2.0000 + 2.0000i   3.0000 + 3.0000i
   4.0000 + 4.0000i   5.0000 + 5.0000i   6.0000 + 6.0000i
F=E'
```

指令运行结果为：

```
F =
   1.0000 - 1.0000i   4.0000 - 4.0000i
   2.0000 - 2.0000i   5.0000 - 5.0000i
   3.0000 - 3.0000i   6.0000 - 6.0000i
```

指令运行结果为：

```
G=E.'
G =
   1.0000 + 1.0000i   4.0000 + 4.0000i
   2.0000 + 2.0000i   5.0000 + 5.0000i
   3.0000 + 3.0000i   6.0000 + 6.0000i
```

运算结果表明，数组的转置运算不同于矩阵的转置运算。

2.3.4 数组的关系运算与逻辑运算及其数组函数

在 MATLAB 中，关系运算与逻辑运算只适用于数组，不适用于矩阵。

2.3.4.1 数组的关系运算

数组的关系运算符已经介绍过，现将其运算规则予以说明：

（1）关系运算的优先级高于算术运算，低于逻辑运算。

（2）运算符 <、<=、>、>= 只比较二标量的实部，而运算符 == 与 ~= 则同时比较二标量的实部与虚部。

（3）若二标量比较，其关系成立者，运算结果为逻辑真（1）；否则为逻辑假（0）。

【例 2-26】数组的关系运算示例。

在 MATLAB 命令窗口输入指令并回车（以下举例同此）。

```
a=[1+2i];b=[3+4i];
a==b
```

当指令运行后可得运算结果逻辑假（0）：

```
ans =
    0
```

（4）若一标量与一数组比较，则将数组与标量的每一个元素逐个比较，其运算结果为一个与数组大小（行列数）相同的数组，其元素由“1”与“0”组成。

【例 2-27】标量与数组比较示例。

```
c=8;d=[4 5 6;7 8 9];
d<c
```

当指令回车后可得运算结果：

```
ans =
    1     1     1
    1     0     0
```

（5）若二数组比较，数组维数大小（行、列数）需相同，将二数组对应的每一个元素逐个比较，其运算结果为一个与比较数组大小相同的数组，其元素由“1”与“0”组成。

【例 2-28】数组与数组比较示例。

```
e=[1 4;8 3];f=[2 6;5 7];
e>f
```

当指令回车后可得运算结果：

```
ans =
    0     0
    1     0
```

2.3.4.2 数组的逻辑运算

数组的逻辑运算符也已经介绍过，现将其运算规则予以说明：

（1）逻辑运算规定：非 0 元素代表逻辑真“1”；0 元素代表逻辑假“0”。

（2）若二标量比较，二标量全为非 0 时，运算结果为“1”，否则为“0”。

（3）逻辑运算中,not 的优先级最高，and 与 or 有相同的优先级（Xor 只有函数形式）：还可用括号改变运算优先权。

（4）若一标量与一数组比较，则将数组与标量的每一个元素逐个比较，其运算结果为一个与数组大小（行列数）相同的数组，其元素由“1”与“0”组成。

【例 2-29】标量与数组的逻辑运算示例。

```
c=6;d=[4 0 6;0 8 0];
d|c
```

当指令回车后可得运算结果：

```
ans =
     1     1     1
     1     1     1
```

（5）若二数组比较，数组维数大小（行、列数）需相同，将二数组对应的每一个元素逐个比较，其运算结果为一个与比较数组大小相同的数组，其元素由“1”与“0”组成。

【例 2-30】数组与数组的逻辑运算示例。

```
e=[1 0;8 3];f=[2 0;5 7];
e&f
```

当指令回车后可得运算结果：

```
ans =
     1     0
     1     1
```

2.3.4.3 数组函数

在 MATLAB 中，对数组运算的函数可以分为基本数组函数与特殊数组函数两类。基本数组函数是指三角函数、对数函数、指数函数、复数运算函数等，这些函数执行时，遵循数组运算规则，即针对数组每个元素进行同等操作。特殊数组函数满足标量运算规则。

MATLAB 的基本数组函数如表 2-9 所示。

表 2-9 MATLAB 的基本数组函数

函数名	功 能	函数名	功 能
sin	正弦	asech	反双曲正割
cos	余弦	acsch	反双曲余割
tan	正切	exp	指数
cot	余切	log	自然对数
set	正割	log10	常用对数
csc	余割	log2	以 2 为底的对数
asin	反正弦	pow2	以 2 为底的指数
acos	反余弦	sqrt	平方根
atan	反正切	abs	绝对值（复数的模）
atan2	四象限反正切	angle	复数的相角
acot	反余切	conj	复数的共轭
asec	反正割	imag	复数的虚部
acsc	反余割	real	复数的实部
sinh	双曲正弦	isreal	是否为复数数组
cosh	双曲余弦	fix	向 0 取整
tanh	双曲正切	floor	向负无穷方向取整
coth	双曲余切	ceil	向正无穷方向取整
asinh	反双曲正弦	round	四舍五入
acosh	反双曲余弦	mod	除法求余（与除数同号）
atanh	反双曲正切	rem	除法求余（与被除数同号）
acoth	反双曲余切	sign	符号函数

关于基本数组函数的使用格式，请参见联机帮助。

MATLAB 的特殊数组函数如表 2-10 所示。

表 2-10　MATLAB 的特殊数组函数

函数名	功　能	函数名	功　能
airy	Airy 函数	erf	误差函数
besselj	第一类 Beseel 函数	erfinv	误差函数的逆函数
bessely	第二类 Beseel 函数	exprint	指数形式积分函数
besselh	第三类 Beseel 函数	gamma	Gamma 函数
beseeli	改进的第一类 Beseel 函数	gammainc	不完全 Gamma 函数
beseelk	改进的第二类 Beseel 函数	gammaln	对数 Gamma 函数
beta	Beta 函数	isprime	是否为质数
betainc	不完全 Beta 函数	gcd	最大公约数
betaln	对数 Beta 函数	lcm	最小公倍数
ellipj	Jacobi 椭圆形函数	rat	有理数近似
ellipke	完全椭圆形积分	rats	有理数形式输出

关于基本数组函数矩的使用格式，请参见联机帮助。

2.3.5　矩阵函数

在 MATLAB 中，有很多种关于矩阵的函数，例如生成矩阵的函数，矩阵变换的函数，矩阵运算的函数等等。

2.3.5.1　矩阵生成函数

MATLAB 提供了一些常用矩阵生成函数。

1．zeros（）函数

zeros（）函数能够生成元素全为 0 的矩阵。

【例 2-31】zeros（）函数应用示例。

```
C=zeros(3,2)
```

当指令回车后可生成如下矩阵：

```
C =
     0     0
     0     0
     0     0
```

2．ones（）函数

ones（）函数能够生成元素全为 1 的矩阵。

【例 2-32】ones（）函数应用示例。

```
D=ones(2,3)
```

当指令回车后可生成矩阵：

```
D =
     1     1     1
     1     1     1
```

3. rand（）函数

rand（）函数能够生成均匀分布随机矩阵。

【例 2-33】rand（）函数应用示例。

```
E=rand(2,5)
```

当指令回车后可生成矩阵 ***E***，其元素数值范围为（0.0,1.0）：

```
E =
    0.9501    0.6068    0.8913    0.4565    0.8214
    0.2311    0.4860    0.7621    0.0185    0.4447
```

4. randn（）函数

randn（）函数能够生成正态分布随机矩阵。

【例 2-34】randn（）函数应用示例。

```
F=randn(3,5)
```

当指令回车后可生成如下矩阵：

```
F =
   -0.4326    0.2877    1.1892    0.1746   -0.5883
   -1.6656   -1.1465   -0.0376   -0.1867    2.1832
    0.1253    1.1909    0.3273    0.7258   -0.1364
```

5. magic（）函数

在 MATLAB 中，magic（）函数是一个特殊的函数，该函数能够自动生成一种特别的 *N* 阶方阵（其中，*N*=1,3,4,5,…），叫做魔方阵。这些 *N* 阶方阵具有一个共同的奇妙特性，即每一行、每一列或对角线上的元素之和都相等。

【例 2-35】magic（）函数应用示例。

```
X=magic(6)
```

当指令回车后可生成如下 6 阶方阵：

```
X =
    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11
```

由 6 阶方阵不难发现，矩阵 ***X*** 的任意行、列和对角线上的元素之和都是 111。使用 magic（）函数产生任意阶魔方阵都是任意行、列和对角线上的元素之和均相等。

6. diag（）函数

diag（）函数可以生成对角矩阵。

【例 2-36】diag（）函数应用示例。

```
Y=magic(4);diag(Y)
```

当指令回车后可生成矩阵 ***X*** 的对角矩阵：

```
ans =
```

```
    16
    11
     6
     1
```

7. triu（）函数

triu（）函数可以生成上三角矩阵。

【例 2-37】triu（）函数应用示例。

```
X=magic(6);triu(X)
```

当指令回车后可生成矩阵 ***X*** 的上三角矩阵：

```
ans =
    35     1     6    26    19    24
     0    32     7    21    23    25
     0     0     2    22    27    20
     0     0     0    17    10    15
     0     0     0     0    14    16
     0     0     0     0     0    11
```

8. tril（）函数

tril（）函数可以生成下三角矩阵。

【例 2-38】tril（）函数应用示例。

```
X=magic(6);tril(X)
```

当指令回车后可生成矩阵 ***X*** 的下三角矩阵：

```
ans =
    35     0     0     0     0     0
     3    32     0     0     0     0
    31     9     2     0     0     0
     8    28    33    17     0     0
    30     5    34    12    14     0
     4    36    29    13    18    11
```

9. size（）函数

size（）函数用来返回指定矩阵的行数和列数。

【例 2-39】size（）函数应用示例。

```
a=[1 2 3;4 5 6;7 8 9];b=size(a)
```

当指令回车后可得矩阵的行数和列数：

```
b =
     3     3
```

10. eye（）函数

eye（）函数可以生成指定行数和列数的单位矩阵。eye（）函数和 size（）函数常常一起使用，用以生成与指定矩阵 ***a*** 同维的单位矩阵。

【例 2-40】eye（）函数应用示例。

```
a=[1 2 3;4 5 6;7 8 9];b=size(a); eye(b)
```

当指令运行后可生成一个 3*3 的单位矩阵：

```
ans =
     1     0     0
     0     1     0
     0     0     1
```

除了以上几个函数外，MATLAB 还提供一些生成特殊矩阵的函数，如表 2-11 所示。

表 2-11　特殊矩阵生成函数表

函数	功 能	函 数	功 能
compan	伴随阵	magic	魔方阵
gallery	Higham 检验矩阵	pascal	Pascal 阵
hadamard	Hadamard 阵	rosser	经典对称特征值检验矩阵
hankel	Hankel 阵	toeplitz	Toeplitz 阵
hilb	Hilbert 阵	vander	Vander 阵
invhilb	逆 Hilbert 阵	wilknsion	wilknsion 特征值检验矩阵

关于特殊矩阵生成函数的使用格式，请参见联机帮助。

2.3.5.2　矩阵变换函数

MATLAB 提供了一些用于矩阵变换的函数，如表 2-12 所示。

表 2-12　矩阵变换函数

函　数	功　能
fliplr	矩阵左右翻转
flipud	矩阵上下翻转
flipdim	矩阵沿特定维翻转
rot90	矩阵逆时针旋转 90°

关于矩阵变换函数的使用格式，请参见联机帮助。

2.3.5.3　矩阵运算函数

MATLAB 还提供了一些用于矩阵运算的函数，如表 2-13 所示。

表 2-13　矩阵运算函数

函　数	功　能	函　数	功　能
cond	矩阵的条件数	norm	向量矩阵的范数
condeig	对应于特征值的条件数	null	矩阵的零空间
det	方阵的行列式值	orth	矩阵的列空间
expm	矩阵的指数函数	rank	矩阵的秩
funm	计算矩阵的函数值	subspace	子空间的夹角
logm	矩阵的对数函数	sqrtm	矩阵的平方根

关于矩阵变换函数的使用格式，请参见联机帮助。

2.3.6　多项式及其运算

在 MATLAB 中采用行向量表示多项式，行向量内存放按降幂次排列的多项式系数。在 MATLAB 中建立多项式就是输入多项式系数行向量。

多项式 $P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$ 的系数行向量为：
$P=[a_0\ a_1\ \cdots\ a_{n-1}\ a_n]$

2.3.6.1 多项式运算函数

MATLAB 提供了用于多项式运算的函数，如表 2-14 所示。

表 2-14 多项式运算函数

函 数	功 能	函 数	功 能
conv	多项式乘法（卷积）	polyval	多项式求值
deconv	多项式除法（解卷）	polyvalm	矩阵多项式求值
poly	由根求多项式	residue	多项式部分分式展开（留数）
polyder	多项式求导数	roots	多项式求根
polyfit	多项式曲线拟合		

2.3.6.2 多项式运算举例

1. 多项式求根

求一个多项式的根（即求多项式的零点）在自动控制的各个领域里是经常遇到的事。

【例 2-41】求多项式 $P(x)=x^4+2x^3+3x^2+4x+5$ 的根。

在 MATLAB 命令窗口输入指令并回车（以下举例同此）。

```
p=[1 2 3 4 5];roots(p)
```

当指令运行后可求得多项式的根：

```
ans =
   0.2878 + 1.4161i
   0.2878 - 1.4161i
  -1.2878 + 0.8579i
  -1.2878 - 0.8579i
```

2. 由根求多项式

【例 2-42】已知多项式的根分别为 1、5、8，试求此根对应的多项式。

```
P1=poly([1,5,8])
```

当指令运行后可求得多项式：

```
P1 =
     1   -14    53   -40
```

即所求多项式为：$P(x)=x^3-14x^2+53x-40$。请注意 poly（）函数的调用格式。所求多项式是否正确，可以用 roots（）函数来验证。

```
p=[1,-14,53,-40];roots(p)
ans =
    8.0000
    5.0000
    1.0000
```

由此可见，多项式的 poly（）与 roots（）函数互为逆运算。

请注意，求矩阵的特征根或特征值，也要用 poly（）函数。

【例 2-43】求矩阵的特征根示例。

```
a=[1 2 3;4 5 6;7 8 9];b=poly(a)
```

当指令运行后可求得矩阵的特征根为：

```
b =
    1.0000  -15.0000  -18.0000   -0.0000
```

3. 多项式求值

【例 2-44】求得多项式在指定点处的值。

```
p=[1 2 3 4 5];polyval(p,3)
```

当指令运行后可求得多项式在 x=3 处的值为：

```
ans =
   179
```

4. 多项式乘法（卷积）与多项式除法（解卷）

【例 2-45】求得多项式的卷积。

```
p=[1 2 3];q=[4 5 6];r=conv(p,q)
```

当指令运行后可求得多项式的卷积：

```
r =
4     13     28     27     18
```

运算结果可以用 deconv（）函数（解卷）来验证：

```
s=deconv(r,p)
s =
    4      5      6
```

由此可见，多项式的 conv（）与 deconv（）函数互为逆运算。

5. 多项式部分分式展开

自动控制的时域分析中，常常要求像函数 $F(s)$ 的反变换。这可以用分解 $F(s)$ 成部分分式的方法，只需要求出部分分式的待定系数就可以了。在自动控制里要利用公式求待定系数，虽然计算不是很难，但是也够繁琐的。而用 MATLAB 的 residue（）函数命令，则多项式部分分式展开的运算极为简单方便。

【例 2-46】已知一传递函数 $F(s)=\dfrac{s+2}{s^2+4s+3}$，将其分解为部分分式。

```
a=[1 2];b=[1 4 3];[z,p,k]=residue(a,b)
```

当指令运行后可求得传递函数的零点、极点与增益：

```
z =
    0.5000
    0.5000
p =
    -3
    -1
```

```
k =
     []
```

即得传递函数分解的部分分式：$F(s)=\frac{0.5}{s+1}+\frac{0.5}{s+3}$。

6．多项式求导数

【例 2-47】求多项式 $P(x)=x^4+2x^3+3x^2+4x+5$ 的导数。

```
a=[1 2 3 4 5];a1=polyder(a)
```

当指令运行后可求得多项式一阶导数多项式系数：

```
a1 =
     4     6     6     4
```

7．多项式曲线拟合

MATLAB 中函数 polyfit（）采用最小二乘法对给定点集进行曲线拟合，并给出拟合多项式系数。

【例 2-48】多项式曲线拟合示例。

```
x=0:0.1:1;y=[2.1 2.3 2.5 2.9 3.2 3.3 3.8 4.1 4.9 5.4 5.8];
p=polyfit(x,y,3)
```

当指令运行后可求得三阶拟合的多项式系数：

```
p =
    0.6993    1.2005    1.8869    2.1077
```

8．矩阵多项式求值

【例 2-49】求矩阵多项式的值。

```
a=[1 2;3 4];p=a^2+3*a+2;polyvalm(p,a)
```

当指令运行后可求得矩阵多项式的值：

```
ans =
    38    24
    36    74
```

2.3.7 字符串

在 MATLAB 中，字符串作为字符数组用单引号引用到程序中。

【例 2-50】

```
c='Ok MATLAB'
```

回车后结果如下：

```
c =
   Ok MATLAB
```

请注意，变量 c 实际上是一个 1*9 字符数组，因为字符数组中空格也算一个字符。

在 MATLAB 里，字符是以 ASCII 数值的格式储存的。用户可以使用如下命令查看变量 c 在 MATLAB 内部的存储格式：

```
x=double(c)
```

```
x =
    79   107    32    77    65    84    76    65    66
```

可以看到，变量c中的每个元素被转化成ASCII码的相应数值。

用户还可以使用函数char（）将ASCII码的相应数字转化还原成字符。例如，输入以下指令：

```
char (x)
```

回车后结果如下：

```
ans =
    Ok MATLAB
```

2.4 M文件与MATLAB函数

M文件是使用MATLAB语言编写的程序代码文件。之所以称为M文件，是因为这种文件都以“.m”作为文件的扩展名。用户可以通过任何文本编辑器或字处理器来生成或编辑M文件，但是在MATLAB提供的编辑/调试器（Editor/Debuger）里生成或编辑M文件最为简单、方便且高效。M文件可以分为两种类型，一种是脚本（或称命令）文件，另一种是函数文件。多个、多种（既有函数文件又有脚本文件的组成成分）M文件组合成为一个MATLAB程序，所以MATLAB程序也是一种类似脚本文件的M文件。

2.4.1 脚本（Script）文件

在MATLAB中，实现某项功能的一串MATLAB命令与函数组合的M文件叫脚本文件。有些MATLAB书籍、文献将“Script file”译为“命令文件”。这种M文件在MATLAB的工作空间内对数据进行操作，能在MATLAB环境下直接执行。脚本文件不仅能够对工作空间内已存在的变量进行操作，并能将建立的变量及其执行后的结果保存在MATLAB的工作空间里，供在以后的计算中被使用。除此之外，脚本文件执行后的结果既可以显示输出，也能够使用MATLAB的绘图函数来产生图形输出结果。

与函数文件相比，脚本文件既不带输入参量也不带输出参量。

2.4.2 函数（Function）文件

函数文件是MATLAB语言最重要的组成部分，MATLAB提供的各种各样的工具箱差不多都是以函数形式给出的。MATLAB的工具箱是内容极为丰富的函数库，可以实现各种各样的功能。这些函数使用时，是作为命令来对待的。故函数有时又称作函数命令。

MATLAB中的函数即为函数文件，是M文件最主要的形式。函数是能够接受输入参数并返回输出参数的M文件。在MATLAB中，函数名和M文件名必须相同。

函数只能在函数体内对变量进行操作，也就是只能访问函数本身工作空间中的变量。

MATLAB中的函数是以两种不同的形式提供给用户的。一种是内置在MATLAB核心中最基本的运算函数，如sqrt（）函数和sin（）函数，其执行效率非常高。另一种是通过M文件实现的函数，如gamma（）函数，这些函数大部分是在最基本运算函数的基

础上扩展而来，扩展的目的是为了实现与完善 MATLAB 新的功能。

2.4.2.1 函数文件的结构

为了说明函数文件的结构，先来看一个 MATLAB 函数文件的例子。

【例 2-51】查看 MATLAB 函数文件“angle.m”的源代码。

在 MATLAB 的命令窗口中输入：

```
type angle.m
```

指令运行后显示函数文件“angle.m”的源代码如下：

```
function p = angle (h)
% ANGLE  Phase angle.
% ANGLE (H)  returns the phase angles, in radians, of a matrix with
% complex elements.
% See also ABS, UNWRAP.
% Copyright  (c)  1984-97 by The MathWorks, Inc.
% $Revision: 5.2 $  $Date: 1997/04/08 05:44:37 $
% Clever way:
% p = imag (log (h)) ;
% Way we'll do it:
p = atan2 (imag (h) , real (h)) ;
```

由此例可见，函数 M 文件一般由 5 个部分组成：

（1）函数定义行；

（2）H1 行；

（3）函数帮助文本；

（4）函数体；

（5）注释。

现将各个组成部分的作用简介如下。

1．函数定义行

函数 M 文件首行总是以关键字“function”开头，并在首行中列出全部输入、输出参量以及函数名。函数名应与对应的 M 文件名相同。输出参量紧跟在“function”之后，常用方括号括起来；输入参量紧跟在函数名之后，常用圆括号括起来。

如果函数有多个输入输出参数，则多个参数之间用逗号分隔。

2．H1 行

H1 行是函数帮助文本的第一行，以“%”号开头，用来概要说明该函数的功能。在 MATLAB 中用命令“lookfor”查找某个函数时，查找到的就是函数 H1 行及其相关信息。

3．函数帮助文本

在 H1 之后而在函数体之前的说明文本就是函数帮助文本。函数帮助文本可以有多行，每行均以“%”号开头，用来比较详细地对该函数进行注释，说明函数的功能与用法。

在 MATLAB 中用命令“help 函数名”联机帮助时，就会显示函数 H1 行与帮助文本

的内容。

4．函数体

函数体是函数的主要部分，是实现该函数功能、进行运算所有程序代码的执行语句。在函数体中，除了进行数值运算外，还包括函数调用与程序调用的必要注释。函数体是函数的精华所在，函数编制水平的高低主要体现在函数体里。函数体中的所有参数都是函数内部变量，这些变量与 MATLAB 工作空间内的变量是分离的。

5．注释

除在 H1 行、函数帮助文本、函数体里以“%”号开头的注释外，有的函数在函数体后还跟注释行，用以说明各种需要交代的问题。以“%”号开头的注释行不论在函数的什么地方，都不是 MATLAB 的执行语句，而仅仅是解释性的说明。

2.4.2.2　函数文件的参数

在函数的首行中必有其输入、输出参数。输入参数提供函数执行的必要条件，输出参数则返回函数执行的结果。

1．参数传递规则

函数输入参数可以用对变量赋值的办法来传递，也可以直接输入数据。而输出参数通常都是通过变量来传递的。函数调用的过程即是参数传递的过程。函数有自己的专用工作空间，函数的输入、输出参量以及所有变量都存放在这里，它与 MATLAB 的工作空间是分开的。函数的变量与 MATLAB 工作空间之间唯一的联系就是函数的输入、输出参数。

脚本文件可以调用函数，函数也可以调用脚本文件，相互之间进行参数传递。

函数文件还可以递归调用，即函数调用自身。递归调用的函数必须能够可靠停止，否则递归调用会成为死循环。

2．固定变量“nargin”与“nargout”

函数空间里有两个固定变量“nargin”与“nargout”，其变量值分别为函数实际输入参数的个数与实际输出参数的个数。这对于正确调用函数，准确进行参数传递是很重要的。

3．固定变量“varargin”与“varargout”

函数空间里还有两个固定变量“varargin”与“varargout”，用来将任意个输入参数传递给函数与返回任意多个输出参数。这对于正确调用函数，准确进行参数传递也是非常重要的。

2.4.3　MATLAB 函数分类

MATLAB 为用户提供了门类多样、内容丰富、功能各异的函数。在 MATLAB5.3 版本的函数库里，不包括 MATLAB 工具箱（函数）就有 21 大类函数，它们的名称与子目录名分别为：

（1）演示举例函数（DEMOS）。

（2）数据分析与傅里叶变换函数（DATAFUN）。

（3）数据类型与结构函数（DATATYPES）。

（4）基本数学函数（ELFUN）。

（5）基本矩阵与数组运算函数（ELMAT）。

（6）双重函数与 ODE（常微分方程）解算器函数（FUNFUN）。

（7）常用命令函数（GENERAL）。

（8）二维图形函数（GRAPH2D）。

（9）三维图形函数（GRAPH3D）。

（10）输入与输出函数（IOFUN）。

（11）MATLAB 程序语言结构函数（LANG）。

（12）矩阵函数（MATFUN）。

（13）运算符与特殊运算符函数（OPS）。

（14）多项式与插值函数（POLYFUN）。

（15）稀疏矩阵函数（SPARFUN）。

（16）专用函数（SPECFUN）。

（17）特殊图形函数（SPECGRAPH）。

（18）字符串函数（STRFUN）。

（19）时间与日期函数（TIMEFUN）。

（20）图形用户界面工具函数（UITOOLS）。

（21）视窗界面操作系统函数（WINFUN）。

这些函数文件子目录所在的路径为：MARLABR11\toolbox\matlab。

需要说明的是，MATLAB5.3 版本的 21 大类函数，对于某些运算对象（例如矩阵），不同类型函数的内容有重复又有交叉，而且与老的 MATLAB 版本相比较，函数内容也产生了很大变化。

2.4.4 MATLAB 函数应用举例

在 MATLAB 程序设计与控制系统的计算、仿真中经常使用的函数（不包括工具箱函数）有以下几种类型：

（1）数据分析函数。

（2）基本数学函数。

（3）专用数学函数。

（4）基本矩阵与数组运算函数。

（5）矩阵函数。

（6）多项式与插值函数。

（7）绘图函数。

由于 MATLAB 中函数的数量特多，在 2.3 节里已经介绍了很多函数的使用方法，以下就经常使用的不同类型函数中再选择几个函数使用的示例予以概要说明。

2.4.4.1 数据分析函数应用举例

MATLAB 提供的数据分析函数包括以下 7 类：

（1）基本运算函数。

（2）有限差分函数。

（3）相关函数。

（4）滤波与卷积函数。

（5）富里哀变换函数。

（6）声音与听觉函数。

（7）听觉文件入口/出口函数。可以在 MATLAB 命令窗口内输入命令“help datafun”，查看数据分析函数的列表。

【例 2-52】求复数矩阵的模与相角。

```
a=[1+2i 3-4i;5+6i 7-8i];
b=abs (a)
c=angle (a)
```

指令执行后可得矩阵的模与相角（单位为弧度）如下：

```
b =
    2.2361    5.0000
    7.8102   10.6301
c =
    1.1071   -0.9273
    0.8761   -0.8520
```

【例 2-53】求数组元素的累计和与累计积。

```
a=[1 3 5 7 9 11]
b=cumsum (a)
c=cumprod (a)
```

指令执行后可得数组元素的累计和与累计积如下：

```
a =
     1      3      5      7      9     11
b =
     1      4      9     16     25     36
c =
     1      3      5    105    945  10395
```

【例 2-54】求质数 1111 分解、生成质数行向量、求质数和与标准差。

```
a=factor (1111)
b=primes (40)
c=sum (b)
d=std (b)
```

指令执行后可得结果如下：

```
a =
    11   101
b =
     2     3     5     7    11    13    17    19    23    29    31    37
c =
```

```
    197
d =
    11.6499
```

【例 2-55】求函数的导数与二重积分。

用 MATLAB 符号运算的函数 diff（）与 int（）来求解，以下指令必须在 MATLAB 中运行。

```
syms('x','y');
f=x*y;
dfdxdy=diff(diff(f,x),y),
s=int(int(f,y,x^2,x),x,0,1)
```

指令执行后可得结果如下：

```
dfdxdy =
        1
s =
        1/24
```

以上 MATLAB 指令中，第 3 行执行情况是：第一次函数 f 对 x 求导，第 4 行执行情况是：其导函数第二次再对 y 求导。第一次函数 f 对指定变量 y 求以为 x 为下限、以 x^2 为上限的定积分，其结果第二次再对 x 求以 0 为上限、以 1 为下限的定积分。

2.4.4.2　基本数学函数应用举例

MATLAB 的基本数学函数主要用来实现简单的数学运算。可以在 MATLAB 命令窗口用命令“help elfun”查看 MATLAB 中的基本函数列表。根据其经常使用的列表，基本数学函数大体可以分为：

（1）三角函数。

（2）指数函数。

（3）复数函数。

（4）舍入函数等。

MATLAB 的数组、矩阵都是以复数作为元素的，不像其他高级语言对复数要事先予以定义，然后才能使用。所以复数是 MATLAB 语言所特有的数值操作元素，有关复数的函数是 MATLAB 所特有的，仅举几例说明。

【例 2-56】求复数的实部与虚部。

在 MATLAB 命令窗口输入下输入指令（以下举例同此）。

```
a=real(8-5i)
b=imag(2-7i)
```

指令执行后结果如下：

```
a =
     8
b =
    -7
```

这两个函数用来求复数的实部与虚部。MATLAB 还提供有关复数的函数能够对复数及复数矩阵进行运算。

【例 2-57】 复数矩阵加法、开方、乘方、乘法运算。

```
c=[1+2i 3+4i;5+6i 7+8i];d=[1-1i 2-2i;3-3i 4-4i];
e=c+d
f=sqrt(c)
g=d^2
h=c*d
```

指令执行后结果如下：

```
e =
   2.0000 + 1.0000i   5.0000 + 2.0000i
   8.0000 + 3.0000i  11.0000 + 4.0000i
f =
   1.2720 + 0.7862i   2.0000 + 1.0000i
   2.5308 + 1.1854i   2.9690 + 1.3472i
g =
   0      -14.0000i    0       -20.0000i
   0      -30.0000i    0       -44.0000i
h =
  24.0000 + 4.0000i  34.0000 + 6.0000i
  56.0000 + 4.0000i  82.0000 + 6.0000i
```

【例 2-58】 舍入函数（朝正无穷大方向取整、朝 0 方向取整、朝负无穷大方向取整、四舍五入取整）运算。

```
a=[ 1.23  -0.123;0.0123 -3.456;-4.567 5.678 ];
b=ceil(a)
c=fix(a)
d=floor(a)
e=round(a)
```

指令执行后结果如下：

```
b =
     2     0
     1    -3
    -4     6
c =
     1     0
     0    -3
    -4     5
d =
     1    -1
```

```
    0    -4
   -5     5
e =
    1     0
    0    -3
   -5     6
```

【例 2-59】取对数运算。

```
a=[128 64+16i];
b=log(a)
c=log2(a)
d=log10(a)
```

指令执行后结果如下：

```
b =
   4.8520   4.1892 + 0.2450i
c =
   7.0000   6.0437 + 0.3534i
d =
   2.1072   1.8193 + 0.1064i
```

这里三个函数分别用于求矩阵中元素以 e 为底的自然对数、以 2 为底的对数与以 10 为底的常用对数。

【例 2-60】求复数矩阵的模、共轭复数矩阵、以 e 为底的指数函数。

```
a=[1-2i 3+4i;5-6i 7+8i];
b=abs(a)
c=conj(a)
e=exp(b)
f=exp(c)
```

指令执行后结果如下：

```
b =
    2.2361    5.0000
    7.8102   10.6301
c =
    1.0000 + 2.0000i   3.0000 - 4.0000i
    5.0000 + 6.0000i   7.0000 - 8.0000i
e =
   1.0e+004 *
   0.0009    0.0148
   0.2466    4.1363
f =
   1.0e+003 *
```

```
   -0.0011 + 0.0025i  -0.0131 + 0.0152i
    0.1425 - 0.0415i  -0.1596 - 1.0850i
```

2.4.4.3 专用函数应用举例

MATLAB 提供的专用数学函数包括以下三类：

（1）特殊数学函数。

（2）数论函数。

（3）坐标变换函数。

几乎全部的 MATLAB 专用数学函数都能进行复数运算。查看 MATLAB 专用函数列表的命令是“help specfun”。

以下的几个例子用于简单说明专用数学函数的用法。

【例 2-61】矩阵的 gammal( )函数、以 2 为底的幂及 airy( )函数运算。

```
a=[1 4;3.7 9];
x=gamma (a)
y=pow2 (a)
z=airy (a)
```

指令执行后结果如下：

```
x =
   1.0e+004 *
   0.0001    0.0006
   0.0004    4.0320
y =
   2.0000   16.0000
  12.9960  512.0000
z =
   0.1353    0.0010
   0.0017    0.0000
```

【例 2-62】 求矩阵的最大公约数、最小公倍数。

```
a=[14 6 3;10 5 18];b=[7 24 6;2 15 9];
c=gcd (a,b)
d=lcm (a,b)
```

指令执行后结果如下：

```
c =
     7     6     3
     2     5     9
d =
    14    24     6
    10    15    18
```

2.4.4.4 基本矩阵与数组运算函数应用举例

MATLAB 提供的基本矩阵与数组运算函数包括以下 5 类：

（1）基本矩阵符函数。

（2）基本数组数据函数。

（3）矩阵运算函数。

（4）特殊变量与常数函数。

（5）特殊矩阵函数。

在 MATLAB 命令窗口内输入命令“help elmat”，可以查看基本矩阵与数组运算函数列表。

以下举几个例子用于说明基本矩阵与数组运算函数的用法。

【例 2-63】矩阵的左右、上下翻转、90° 旋转。

```
a=[1 2 3;4 5 6;7 8 9]
b=fliplr(a)
c=flipud(a)
d=rot90(a)
```

指令执行后结果如下：

```
a =
     1     2     3
     4     5     6
     7     8     9
b =
     3     2     1
     6     5     4
     9     8     7
c =
     7     8     9
     4     5     6
     1     2     3
d =
     3     6     9
     2     5     8
     1     4     7
```

【例 2-64】求矩阵的上 Hessenberg 标准形与酉相似三角形。

```
a=[1 2 3;4 5 6;7 8 9];
[p,h]=hess(a)
[u,t]=schur(a)
```

指令执行后结果如下：

```
p =
```

```
    1.0000         0         0
         0   -0.4961   -0.8682
         0   -0.8682    0.4961
h =
    1.0000   -3.5970   -0.2481
   -8.0623   14.0462    2.8308
         0    0.8308   -0.0462
u =
    0.2320    0.8829    0.4082
    0.5253    0.2395   -0.8165
    0.8187   -0.4039    0.4082
t =
   16.1168    4.8990    0.0000
    0.0000   -1.1168   -0.0000
         0         0   -0.0000
```

验证矩阵的上 Hessenberg 标准形计算是否正确。

```
a1=p*h*p'
a2=p'*p
a1 =
    1.0000    2.0000    3.0000
    4.0000    5.0000    6.0000
    7.0000    8.0000    9.0000
a2 =
    1.0000         0         0
         0    1.0000    0.0000
         0    0.0000    1.0000
```

验证矩阵的酉相似三角形计算是否正确。

```
a3=u*t*u'
a4=u'*u
a3 =
    1.0000    2.0000    3.0000
    4.0000    5.0000    6.0000
    7.0000    8.0000    9.0000
a4 =
    1.0000    0.0000   -0.0000
    0.0000    1.0000   -0.0000
   -0.0000   -0.0000    1.0000
```

验算结果说明矩阵的上 Hessenberg 标准形与酉相似三角形计算正确。

【例 2-65】生成 Pascal 、magic、hilb 矩阵。

```
a=pascal(4)
b=magic(4)
c=hilb(4)
```

指令执行后结果如下：

```
a =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
b =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
c =
    1.0000    0.5000    0.3333    0.2500
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429
```

2.4.4.5 矩阵函数应用举例

MATLAB 中有以下 4 类矩阵函数：

（1）矩阵分析函数。

（2）线性方程函数。

（3）特征值与奇异值函数。

（4）因式分解应用函数。

用户可以在命令窗口内输入命令“help matfun”，来查看 MATLAB 矩阵函数列表。

以下举几个例子用于说明矩阵函数的用法。

【例 2-66】求矩阵与逆相关的条件数、矩阵行列式的值、矩阵的秩、矩阵的逆、矩阵的 Hessenberg 形式。

```
a=magic(5);
b=cond(a)
c=det(a)
d=rank(a)
e=inv(a)
h=hess(a)
```

指令组回车执行后结果如下：

矩阵与逆相关的条件数为：b = 5.4618

矩阵行列式的值为：c = 5070000

矩阵的秩为：d = 5

矩阵的逆为：

```
e =
   -0.0049    0.0512   -0.0354    0.0012    0.0034
    0.0431   -0.0373   -0.0046    0.0127    0.0015
   -0.0303    0.0031    0.0031    0.0031    0.0364
    0.0047   -0.0065    0.0108    0.0435   -0.0370
    0.0028    0.0050    0.0415   -0.0450    0.0111
```

矩阵的 Hessenberg 形式为：

```
h =
   17.0000  -28.9413    1.8470   -4.4603    2.2572
  -27.6767   33.9399   26.1875   -2.2280    1.2675
         0   25.0964   20.6871   -6.6055   -0.1973
         0         0   -5.9630  -16.8163  -12.4454
         0         0         0   -9.0122   10.1893
```

【例 2-67】由生成的均匀分布随机矩阵与正态分布随机矩阵求矩阵的广义逆矩阵。

```
a=fix (10*rand (3,3))
b=floor (10*randn (3,3))
e=kron (a,b)
c=round (10*rand (2,3))
d=pinv (c)
```

指令组回车执行后结果如下：

均匀分布随机矩阵 ***a*** 为：

```
a =
     9     4     3
     7     2     9
     4     6     7
```

正态分布随机矩阵 ***b*** 为：

```
b =
    -1   -14    -7
    -9     7     8
     2    16    12
```

矩阵的广义逆矩阵 ***e*** 为：

```
e =
    -9  -126   -63    -4   -56   -28    -3   -42   -21
   -81    63    72   -36    28    32   -27    21    24
    18   144   108     8    64    48     6    48    36
    -7   -98   -49    -2   -28   -14    -9  -126   -63
   -63    49    56   -18    14    16   -81    63    72
```

```
    14   112    84     4    32    24    18   144   108
    -4   -56   -28    -6   -84   -42    -7   -98   -49
   -36    28    32   -54    42    48   -63    49    56
     8    64    48    12    96    72    14   112    84
```

均匀分布随机矩阵 *c* 为：

```
c =
    10     6     9
     2     5     8
```

正态分布随机矩阵 ***d*** 为：

```
d =
 0.1295   -0.1484
-0.0098    0.0666
-0.0262    0.1204
```

【例 2-68】求 3 阶帕斯卡矩阵的 Cholesky 分解 R 并验算。

```
a=pascal (3)
r=chol (a)
```

指令组回车执行后结果如下：

```
a =
     1     1     1
     1     2     3
     1     3     6
r =
     1     1     1
     0     1     2
     0     0     1
```

验算矩阵 ***A*** 的 Cholesky 分解 R 计算是否正确：

```
r'*r
ans =
    1     1     1
    1     2     3
    1     3     6
```

在线性方程函数中，最基本的三角分解函数是将一个方阵表示成两个基本三角阵的积，其中一个三角阵为上三角矩阵，另一个为下三角矩阵，这种分解被称为“lu”分解，“l”表示下三角矩阵，“u”表示上三角矩阵，算法是高斯变量消去法。用户可以使用该运算的逆运算来证明“lu”分解的正确性。

【例 2-69】求矩阵 ***a*** 的 lu 分解并验算。

```
a=[1 2 3;4 5 6;7 8 9];
[l,u]=lu (a)
```

指令组回车执行后结果如下：

```
l =
      0000         0          0
    0.1429    1.0000          0
    0.5714    0.5000     1.0000
u =
    7.0000    8.0000     9.0000
         0    0.8571     1.7143
         0         0     0.0000
```

验算矩阵 ***a*** 的 lu 分解计算是否正确：

```
l*u
ans =
     1     2     3
     4     5     6
     7     8     9
```

在 MATLAB 中，矩阵的正交变换可通过“qr（ ）”函数方便地实现。

【例 2-70】求矩阵 ***b*** 的 qr 分解并验算。

```
b=[9 4;2 8;6 7];
[q,r]=qr (b)
```

指令执行后得分解的正交矩阵与分解的上三角形矩阵分别如下：

```
q =
   -0.8182    0.3999   -0.4131
   -0.1818   -0.8616   -0.4739
   -0.5455   -0.3126    0.7777
r =
  -11.0000   -8.5455
         0   -7.4817
         0         0
```

验算矩阵 ***b*** 的 qr 分解的正确性：

```
q*r
ans =
    9.0000    4.0000
    2.0000    8.0000
    6.0000    7.0000
```

在 MATLAB 中，对矩阵进行奇异值分解是使用“svd（ ）”函数来完成的。奇异值分解后产生三个因数矩阵“u”，“s”，“v”。其中“u”矩阵和“v” 矩阵是正交矩阵，“s”矩阵是对角矩阵，对于被进行奇异值分解的矩阵 ***a***，则 svd（***a***）恰好返回矩阵“s”的对角元素，即 ***a*** 的奇异值。

【例 2-71】对矩阵进行奇异值分解。

```
a=ceil (10*rand (4,4))
```

```
b=svd (a)
[u,s,v]=svd (a)
```

指令执行后结果如下：

```
a =
     8     9     3     7
     9     3     3     2
     3     9     1     9
     3    10     1     2
b =
   22.5851
    7.7397
    4.6854
    0.2356
u =
    0.6273   -0.1731   -0.1463   -0.7451
    0.3556   -0.8012    0.0252    0.4806
    0.5453    0.4929   -0.5115    0.4450
    0.4274    0.2918    0.8464    0.1258
s =
   22.5851         0         0         0
         0    7.7397         0         0
         0         0    4.6854         0
         0         0         0    0.2356
v =
    0.4931   -0.8064    0.0130    0.3262
    0.7037    0.4384    0.5591   -0.0024
    0.1736   -0.2763   -0.0061   -0.9453
    0.4811    0.2850   -0.8290    0.0104
```

对于 n 阶矩阵 A，若有 n 个 λ 值满足 $Ax=\lambda x$，则 λ 为 A 的特征值，x 为 A 的特征向量。MATLAB 中使用函数“eig（*A*）”计算特征值，计算返回的特征值是以列向量的形式给出的。如果矩阵 A 为实对称矩阵，其特征值为实数；如果矩阵 A 为非对称矩阵，其特征值为复数。

【例 2-72】求矩阵的特征值与特征向量

```
a=ceil (10*rand (4,4))
b=eig (a)
[v,d]=eig (a)
```

指令组回车执行后结果如下：

```
a =
     5     5    10     7
```

```
    9     5     1     7
    5     5     3    10
    9     5     1     6
```

a 矩阵的特征值列向量为 ***b***

```
b =
    23.3819
    -2.0780 + 4.7117i
    -2.0780 - 4.7117i
    -0.2258
```

a 矩阵的右特征向量矩阵为 ***v***

```
v =
   0.5689   0.6385 - 0.0847i   0.6385 + 0.0847i   0.3091
   0.4802  -0.3867 - 0.0706i  -0.3867 + 0.0706i  -0.9072
   0.4833   0.0544 + 0.5023i   0.0544 - 0.5023i   0.1069
   0.4605  -0.3903 - 0.1518i  -0.3903 + 0.1518i   0.2646
```

a 矩阵的特征值对角阵为 ***d***

```
d =
   23.3819            0                    0                    0
         0     -2.0780 + 4.7117i           0                    0
         0            0             -2.0780 - 4.7117i           0
         0            0                    0              -0.2258
```

所求出的 ***a*** 矩阵右特征向量矩阵 ***v*** 与特征值对角阵 ***d*** 满足关系：***a*v=v*d***。

```
a*v (或 v*d)
ans =
  13.3028  -0.9280 + 3.1844i  -0.9280 - 3.1844i  -0.0698
  11.2286   1.1360 - 1.6751i   1.1360 + 1.6751i   0.2048
  11.3011  -2.4799 - 0.7873i  -2.4799 + 0.7873i  -0.0241
  10.7681   1.5263 - 1.5233i   1.5263 + 1.5233i  -0.0597
```

2.4.4.6 多项式与插值函数数应用举例

MATLAB 提供的多项式与插值函数包括以下 4 类：

（1）数据插值函数；

（2）线性插值函数；

（3）几何分析函数；

（4） 多项式函数。

在 MATLAB 命令窗口内使用命令“help polyfun”，可查看多项式与插值函数列表。

有关多项式的求根、由根求多项式、多项式求值、多项式乘法（卷积）与多项式除法（解卷）、多项式部分分式展开（留数）、多项式求导数、多项式曲线拟合及矩阵多项式求值等已在 2.3.6 节举例中说明了有关函数的用法，以下再举几个例子说明多项式与

插值函数的用法。

【例 2-73】求多项式$a(x)=x^3+3x^2+5x+7$与$b(x)=2x^2+4x+6$之积与比（a/b）的导数。

```
a=[1 3 5 7];b=[2 4 6];
c=polyder (a,b)
[q,d]=polyder (a,b)
```

当指令组运行后可求得两多项式之积与比的导数：

```
c =
    10    40    84   104    58
q =
     2     8    20     8     2
d =
     4    16    40    48    36
```

【例 2-74】求正态分布矩阵的多项式系数、矩阵多项式的值。

```
a=round (10*randn (3,3))
b=poly (a)
c=poly2str (b,'a')
d=polyvalm (b,a)
```

当指令运行后可求得正态分布矩阵 $\boldsymbol{a}$ 为：

```
a =
    -4     3    12
   -17   -11     0
     1    12     3
```

矩阵的多项式系数 b 为：

```
b =
  1.0e+003 *
    0.0010    0.0120    0.0380    2.0310
```

以习惯的方式显示多项式 c 为：

```
c =
   a^3 + 12 a^2 + 38 a + 2031
```

矩阵多项式的值 d 为：

```
d =
  1.0e-011 *
   -0.3865   -0.0227   -0.0909
    0.1251   -0.3183         0
   -0.0114   -0.0853   -0.4320
```

【例 2-75】对于余弦函数求其线性、三次样条、三次多项式的一元插值。

根据题意，给出调用一元插值函数 interp1（）的程序 L2401.m 如下：

```
% MATLAB PROGRAM L2401.m
%
t=0:2*pi;y=cos(t);ti=0:0.1*pi:2*pi;
li=interp1(t,y,ti);
si=interp1(t,y,ti,'spline');
ci=interp1(t,y,ti,'cubic');
plot(t,y,'o',ti,li,'b',ti,ci,'k:',ti,si,'r')
legend('原数据','linear','cubic','spline')
```

程序运行后可得余弦函数之线性、三次样条、三次多项式的一元插值曲线如图 2-33 所示。

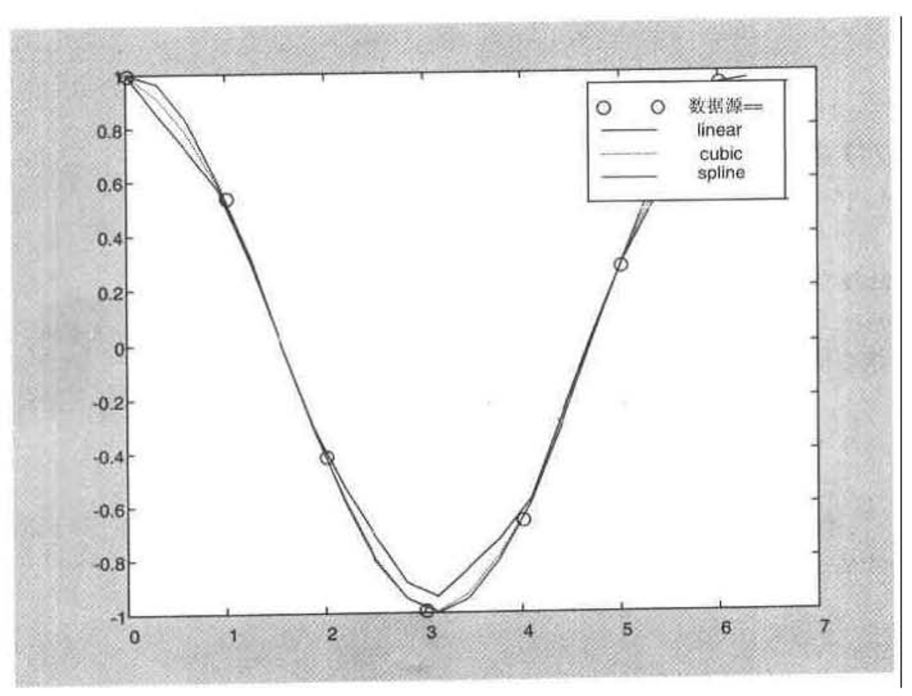


图 2-33　余弦函数的一元插值算法比较

【例 2-76】采用插值方式绘制海底形状图。

根据题意，可先对数据采集范围内线性等分 x、y 轴，产生格点矩阵；其次进行二维插值；最后绘制三维着色图。依照这些考虑，给出如下程序 L2402.m。

```
% MATLAB PROGRAM L2402.m
%
xi=linspace(-5,5,50);yi=linspace(-5,5,50);
[XI,YI]=meshgrid(xi,yi);
ZI=interp2(x,y,z,XI,YI,'*cubic');
surf(XI,YI,ZI),view(-25,25)
```

程序运行后可得海底形状三维着色图，如图 2-34 所示。

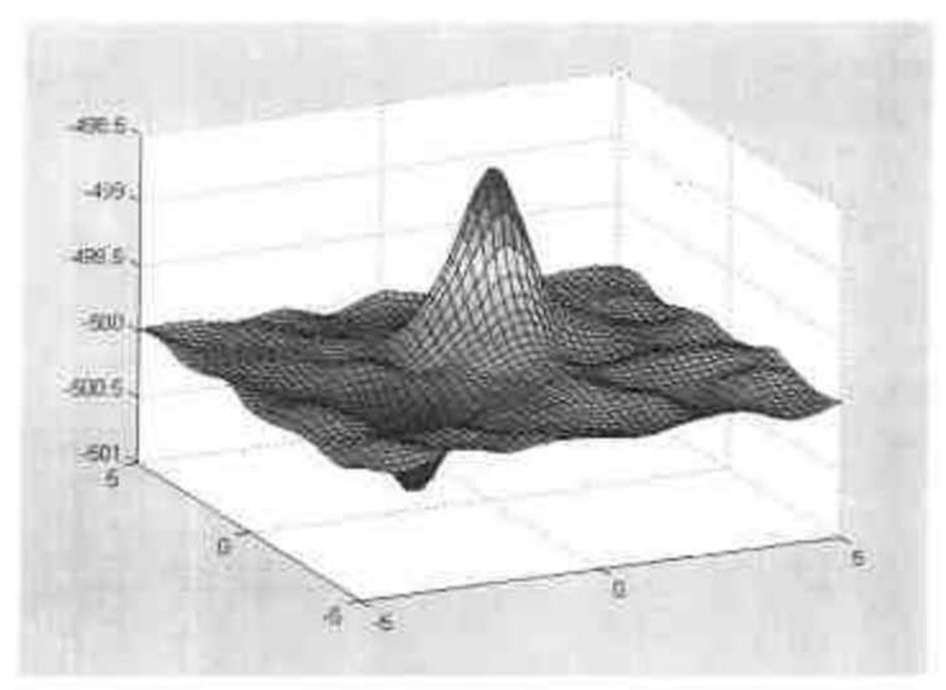


图2-34 海底形状三维着色图

2.4.4.7 绘图函数应用举例

MATLAB的绘图功能特别强。MATLAB提供了三大类绘图函数：

（1）.绘制二维图形函数。

（2）绘制三维图形函数。

（3）特殊图形函数。

可以在MATLAB命令窗口内分别用命令“help graph2d”、“help graph3d”与“help specgraph”，来查看其函数列表。

（1）绘制二维图形函数

绘制二维图形的函数包括以下几个类型：

①绘制X-Y坐标图函数。

②坐标控制函数。

③图形添加注解函数。

④复制与打印图形函数。

MATLAB能够绘制线性坐标、全对数坐标、半对数坐标和极坐标下的二维图形，并能够给图形添加标注。

MTLAB最基本的绘图函数是polt（）。

【例2-77】利用polt（）与semilogx（）函数在同一图形窗口里绘制余弦曲线。

```
% MATLAB PROGRAM L2403.m
%
t=pi:0.1:3*pi;
```

```
y=cos(t);
subplot(1,2,1)
plot(t,y)
xlabel('Time(0.1~3\pi)','FontWeight','bold')
ylabel('Cos(t)','FontWeight','bold')
title('余弦曲线','FontSize',12,'FontName','黑体')
subplot(1,2,2)
semilogx(t,y)
grid on
```

程序运行后可得余弦曲线如图 2-35 所示。

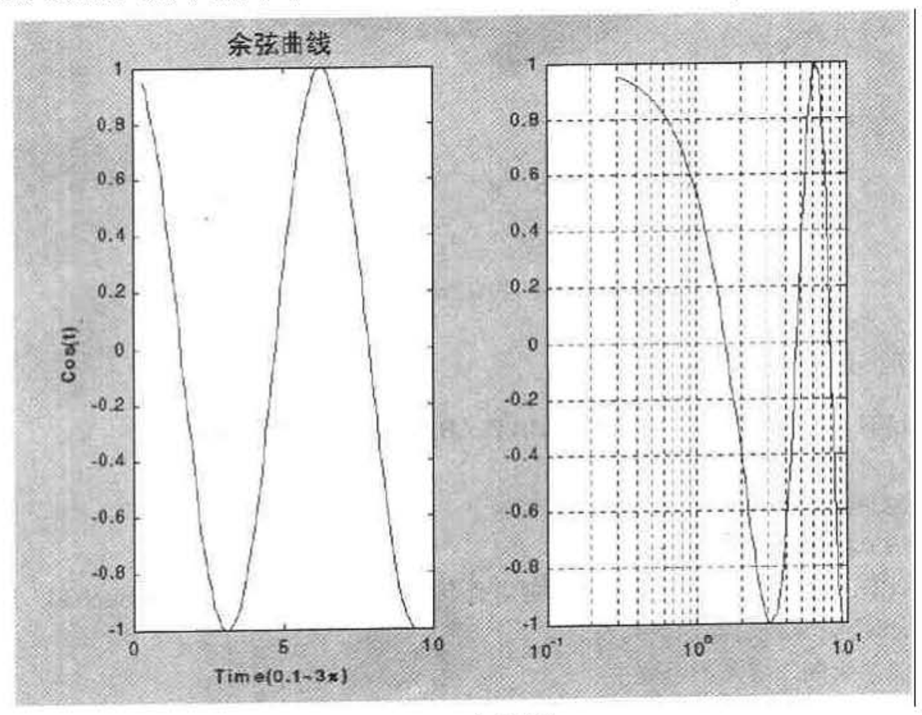


图 2-35　余弦曲线

【例 2-78】绘制心形线与三叶玫瑰线。

根据题意，可以先计算出极坐标图的角度向量，再求极坐标图的幅值向量，最后绘制极坐标图。依照这些考虑，给出调用函数 polar（）的程序 L2404.m 如下。

```
% MATLAB PROGRAM L2404.m
%
a1=1;thita1=0:0.1:2*pi;
rho1=a1*(1-cos(thita1));polar(thita1,rho1)
a2=1;thita2=0:0.1:2*pi;
rho2=a2*cos(3*thita2); polar(thita2,rho2)
```

执行程序后可得如图 2-36 所示的心形线与如图 2-37 所示的三叶玫瑰线。

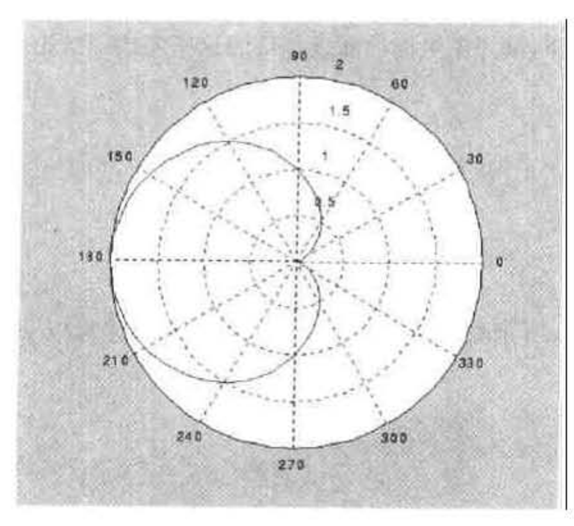


图 2-36　心形线

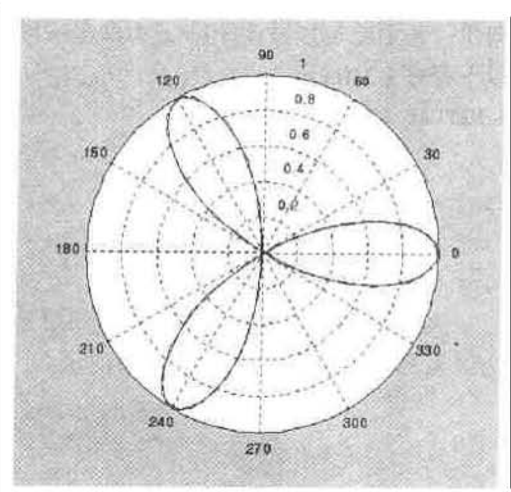


图 2-37　三叶玫瑰线

【例 2-79】调用函数 polotyy（）绘制双 y 轴图形。

```
t=0:0.1:2*pi;
x=sin(t);
y=cos(t);
plotyy(t,y,x+pi,y)
```

执行这一组命令组后可得双 y 轴图，如图 2-38 所示。

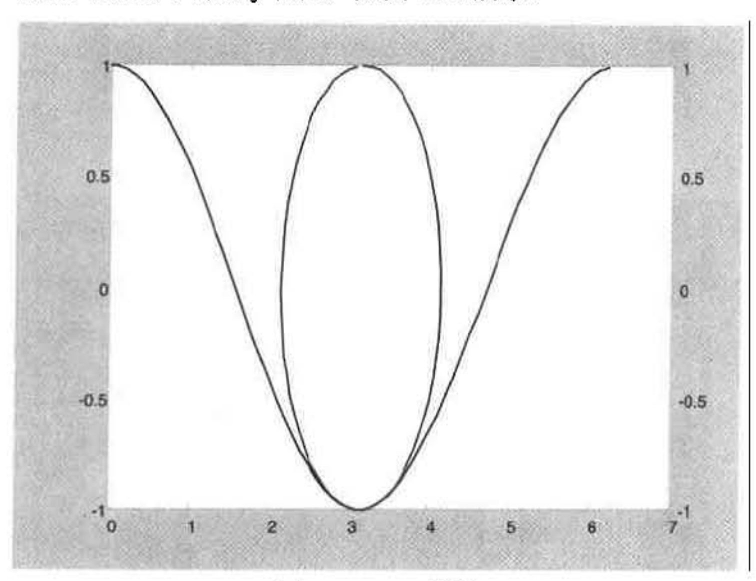


图 2-38　双 y 轴图

图 2-38 中，横坐标为 t，左侧的 y 轴是线性坐标，对应着 cos（t）曲线，右侧的纵轴也是线性坐标，对应着的是以（x+pi）为横坐标的 y 值。

【例 2-80】利用 MATLAB 的 Demos 显示示例函数 peaks（5）绘制其数值矩阵曲线。

根据题意，可先生成 peaks（5）的数值矩阵 x，将其绘制成 5 条曲线；尔后生成 y

向量；最后绘制以数值矩阵 *x* 为横坐标以 *y* 向量为纵坐标的曲线。依照这些考虑，给出以下程序 L2405.m。

```
% MATLAB PROGRAM L2405.m
%
figure (1) ; x=peaks (5) ;plot (x) ;hold on
figure (2) ;y=1:length (peaks (5)) ;
plot (peaks (5) ,y)
```

执行程序后可得数值矩阵曲线如图 2-39 所示与以 peaks（5）数值矩阵为横坐标以 *y* 为纵坐标的 plot 曲线，如图 2-40 所示。

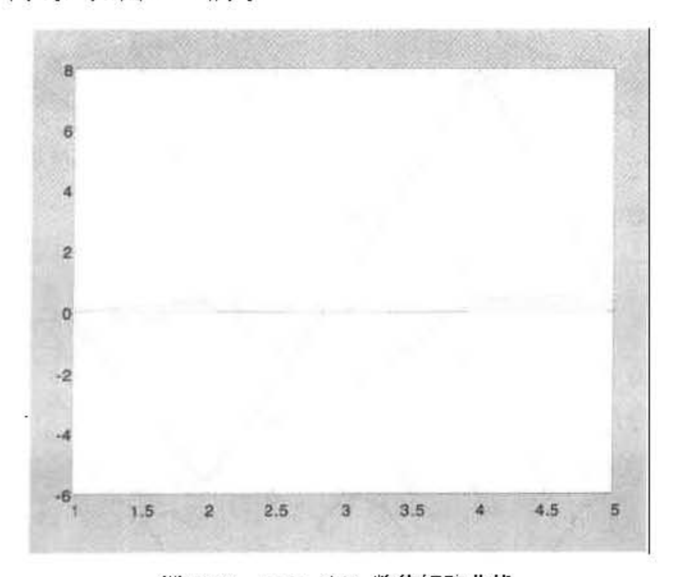


图 2-39　peaks（5）数值矩阵曲线

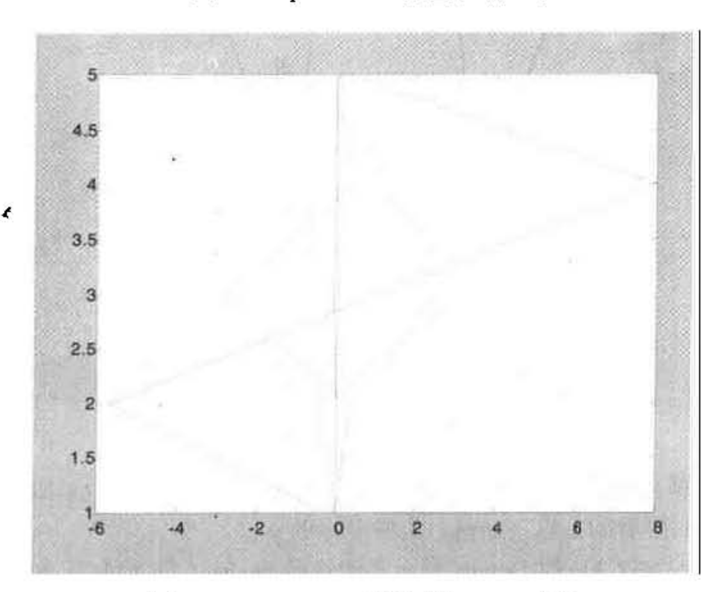


图 2-40　peaks（5）数值矩阵（x,y）曲线

（2）绘制三维图形函数

MATLAB 提供绘制三维图形的函数，使得其绘图功能可以应用到三维空间。用于绘制三维图形的函数包括以下几个类型：

①绘制基本三维图形的函数；

②颜色控制函数；

③光照函数；

④颜色映象函数；

⑤视点控制函数；

⑥坐标轴控制函数；

⑦照相控制函数；

⑧高水平照相控制；

⑨高水平亮度控制；

⑩图形添加注解的函数；

⑪复制与打印图形的函数。

【例 2-81】绘制经典螺旋线。

给出以下指令组来绘制经典螺旋线。

```
t=0:0.1:5*pi;plot3 (sin (t) ,cos (t) ,t)
xlabel ('sin (t) ','FontWeight','bold','FontAngle','italic') ;
ylabel ('cos (t) ','FontWeight','bold','FontAngle','italic') ;
zlabel ('t','FontWeight','bold','FontAngle','italic') ;
```

执行这一指令组后可得到一经典螺旋线如图 2-41 所示。

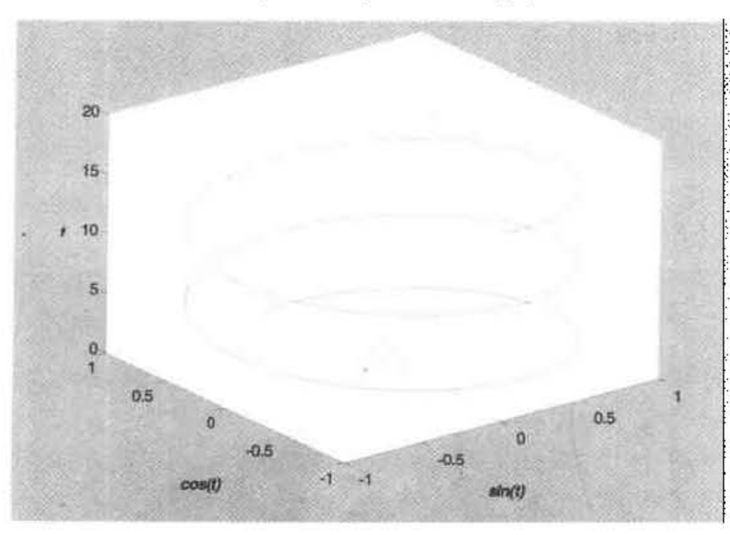


图 2-41　经典螺旋线

【例 2-82】按某给定函数关系绘制由矩阵元素作为参数的三维图。

根据题意，可先从向量生成方阵，再从矩阵 ***x,y*** 按给定函数关系生成矩阵 ***z***，最后作 ***x,y,z*** 的三维图。依照这些考虑，给出以下指令组来绘制由矩阵元素作参数的三维图。

```
[x,y]=meshgrid(-pi:0.05*pi:pi);
```

```
z=sin(x).^3-cos(y);plot3(x,y,z)
```

执行这一命令组后可得一由矩阵元素作参数的三维图如图 2-42 所示。

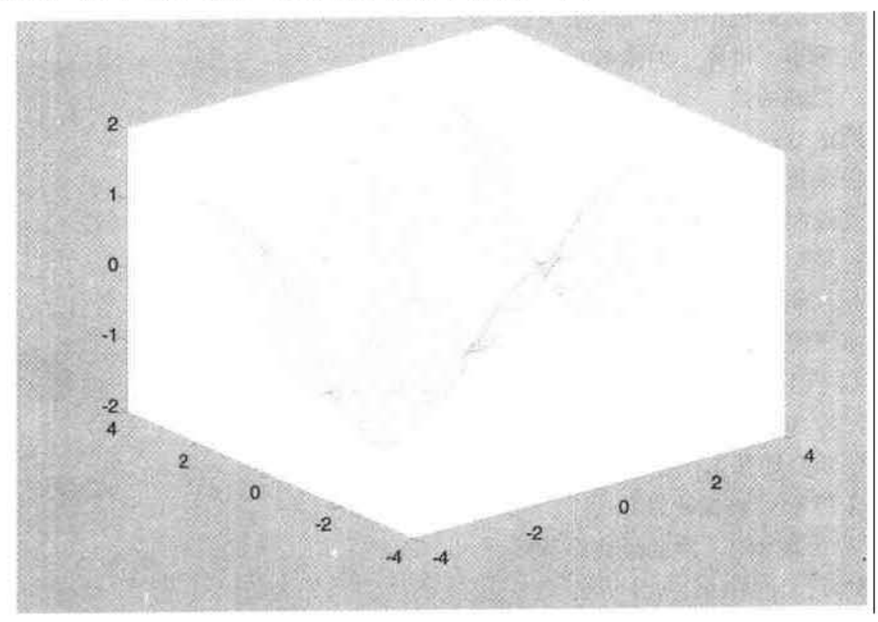


图 2-42　矩阵元素作参数的三维图形

【例 2-83】绘制草帽图。

给出以下指令组绘制草帽图。

```
[x,y]=meshgrid(-8:0.5:8);
r=sqrt(x.^2+y.^2)+eps;
z=sin(r)./r;
surf(x,y,z)
shading interp
axis off
```

执行这一指令组后可得到一草帽图，如图 2-43 所示。

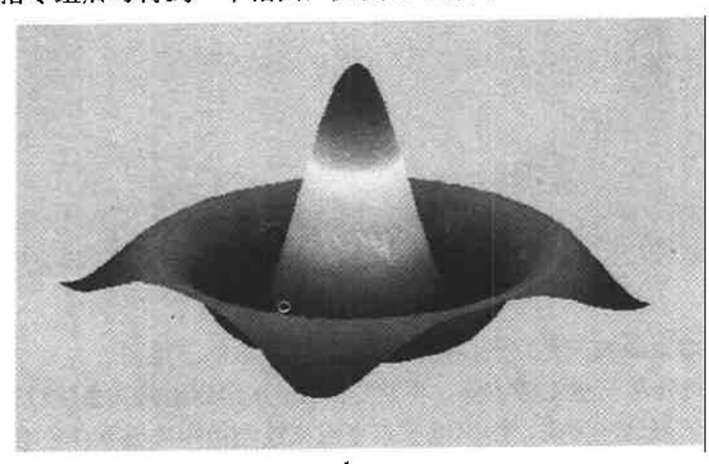


图 2-43　草帽图

（3）特殊图形函数

MATLAB 用于绘制特殊图形的特殊绘图函数包括以下几种类型：

①特殊二维图形函数。

②等高线与 2-1/2 维图形函数。

③特殊三维图形函数。

④容量与矢量视觉函数。

⑤图像显示与 I/O 文件。

⑥产生动画效果函数。

⑦色彩相关函数。

⑧实体模型函数。

MATLAB 还可以用不同的方法绘制特殊图形，如填充图、条形图及饼状图等。

【例 2-84】绘制圆球体图。

给出以下命令组绘制圆球体图。

```
sphere (100) ;axis equal;
shading flat;camlight right;
camlight left;lighting phong
```

执行这一命令组后可得一圆球体，如图 2-44 所示。

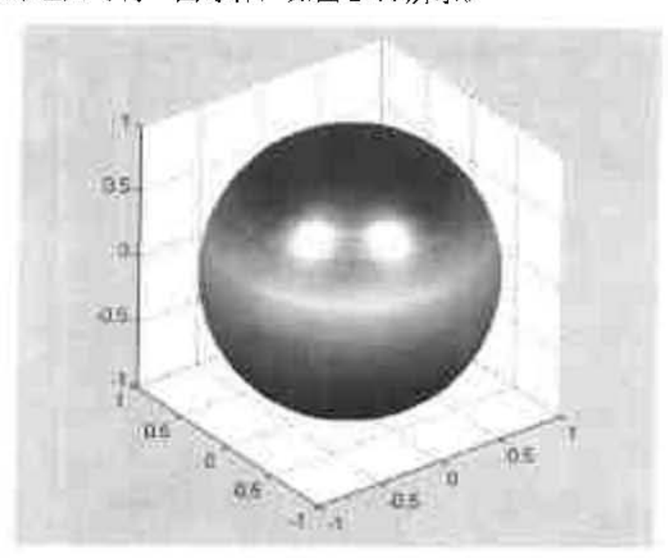


图 2-44　圆球体

【例 2-85】生成 MATLAB 的 Demos 中三维高斯曲面函数 peaks（4）与 eye （17）的三维图形。

在 MATLAB 命令窗中输入以下指令后并回车（以下举例同此）

```
peaks (4)
```

当指令运行后可得函数 peaks（4）三维曲面图，如图 2-45 所示。

```
z=eye (17) ;
mesh (z)
```

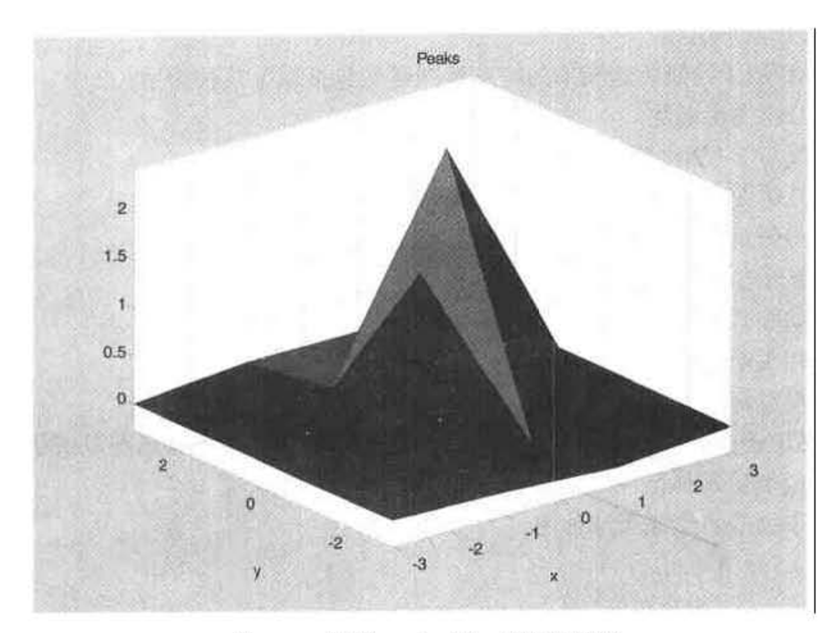


图 2-45　函数 peaks（5）三维曲面图

当指令运行后可得函数 mesh（z）三维图，如图 2-46 所示。

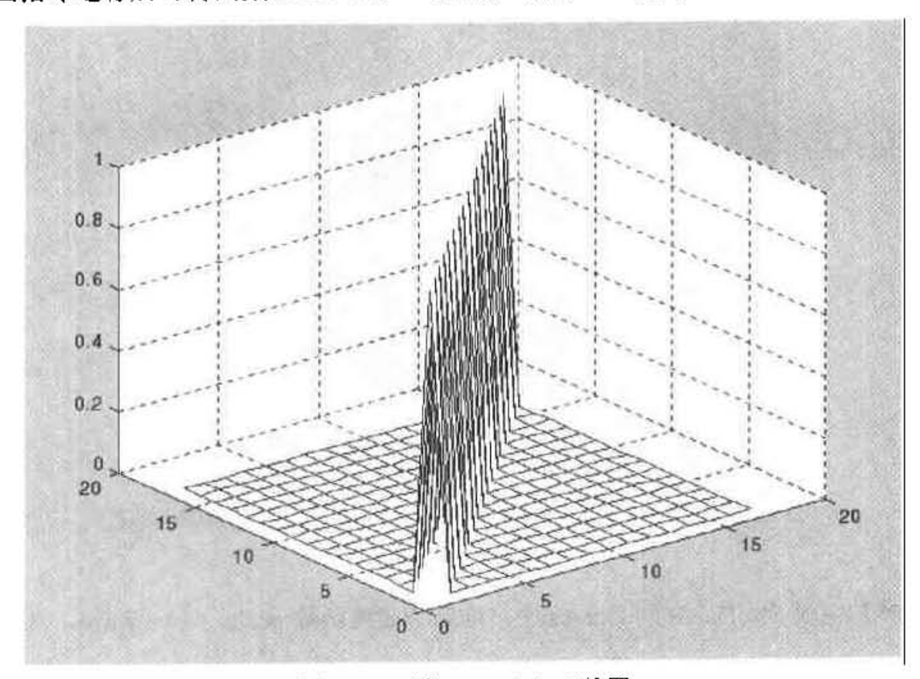


图 2-46　函数 mesh（z）三维图

【例 2-86】绘制均匀分布随机数、正态分布随机数的直方图。

```
y1=rand(10000,2);
hist(y1,20)
```

执行这一指令组后可得两列均匀分布随机数列直方图，如图 2-47 所示。

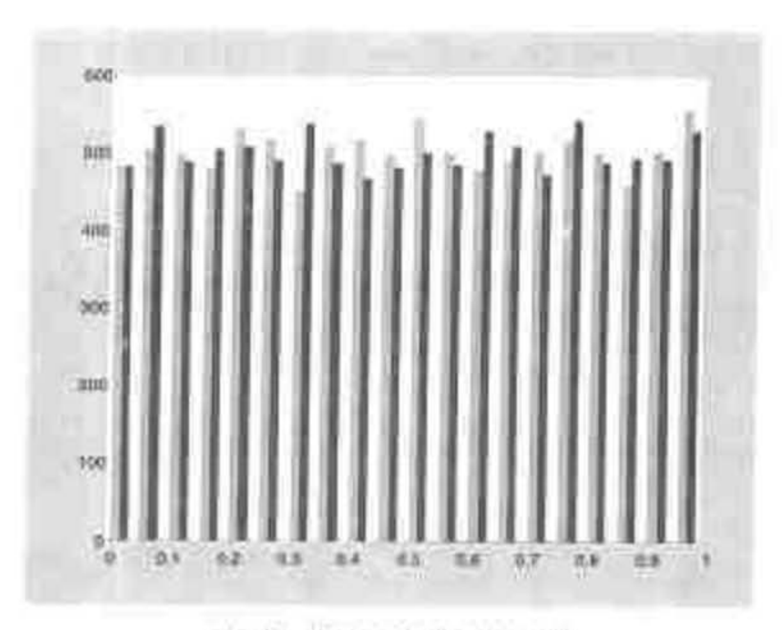


图 2-47 均匀分布随机数列直方图

```
y2=randn (10000,2) ;
hist (y2,30)
```

这组指令执行后可得两列正态分布随机数列直方图，如图 2-48 所示。

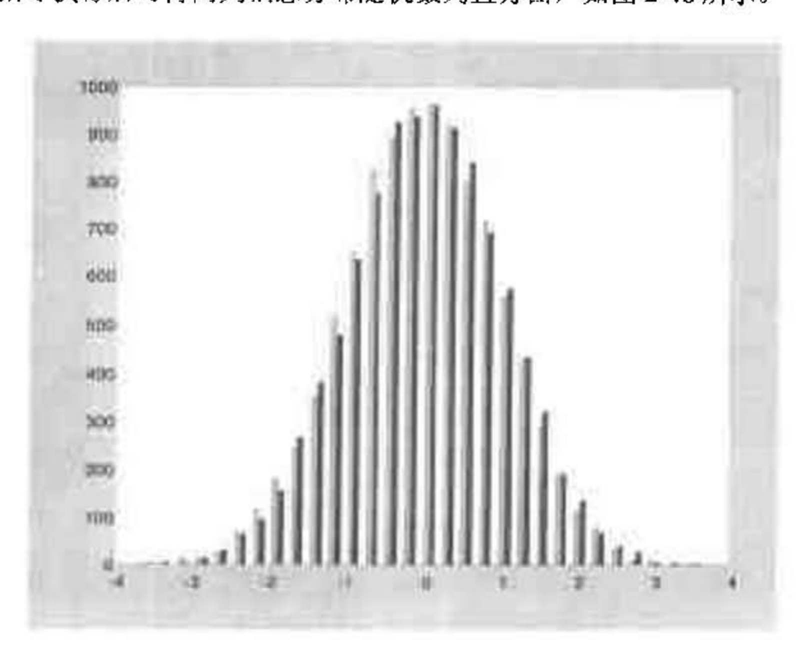


图 2-48 正态分布随机数列直方图

【例 2-87】绘制三维高斯函数 peaks（4）的等高线图。

```
[x,y,z]=peaks (4) ;contour (x,y,z,10)
[x,y,z]=peaks (4) ;contour3 (z,10)
```

这组一指令执行后可得三维高斯函数 peaks（4）的二维等高线图（图 2-49）与三维等高线图（图 2-50）。

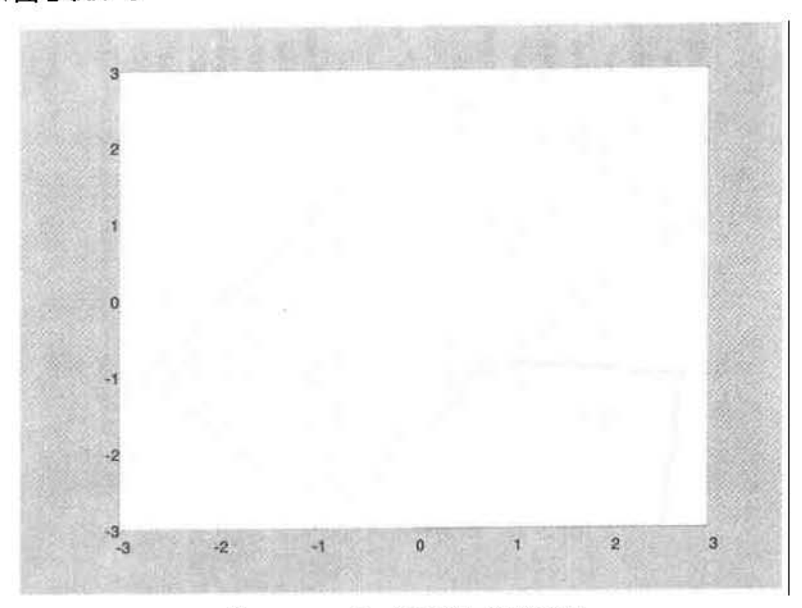


图 2-49　peaks（4）的二维等高线

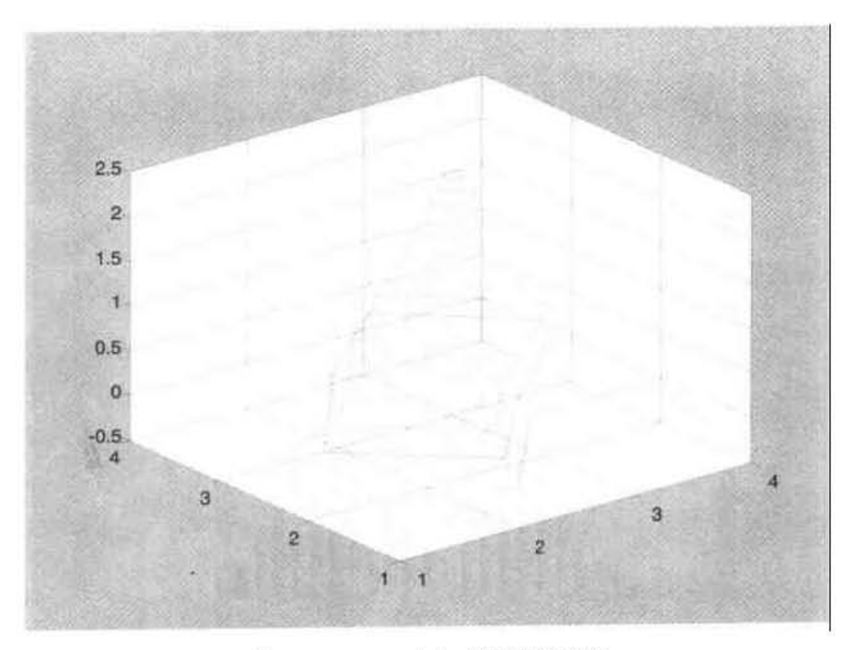


图 2-50　peaks（4）的三维等高线

```
[x,y,z]=peaks (4) ;[c,h]=contourf (z) ;clabel (c,h)
[x,y,z]=peaks (4) ;contourf (x,y,z,10)
```

这组指令执行后可得如图 2-51 所示的三维高斯函数 peaks（4）标注高度值的二维等高线图与如图 2-52 所示的填充二维等高线图。

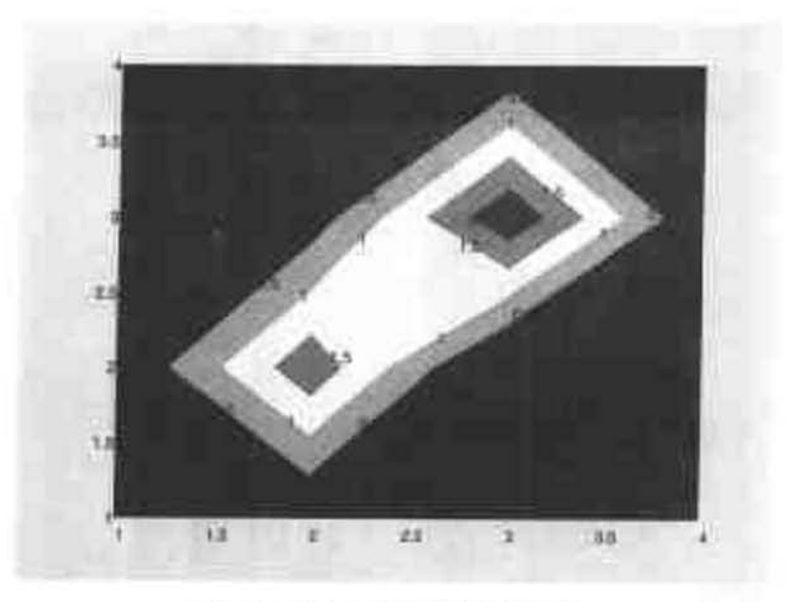


图 2-51 标注高度值的二维等高线图

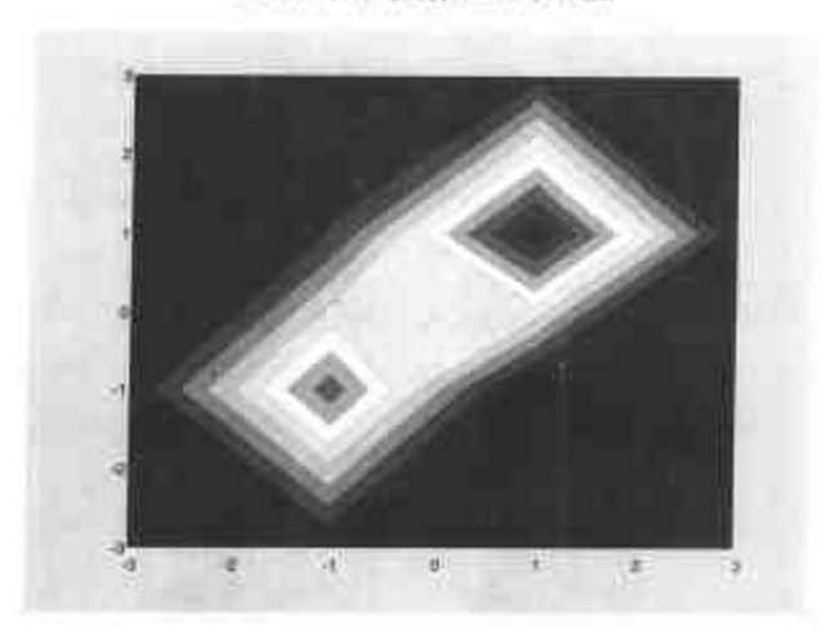


图 2-52 填充的二维等高线图

【例 2-88】绘制矩阵的填充图、条形图。

```
a=[1 2 3;4 5 6;7 8 9];b=[65 78 89; 97 94 90; 86 73 61];
subplot (1,2,1)
area (a)
subplot (1,2,2)
bar (b)
c=[45 89 97 43 61];
```

执行这一指令组后可得矩阵的填充图、条形图，如图 2-53 所示。

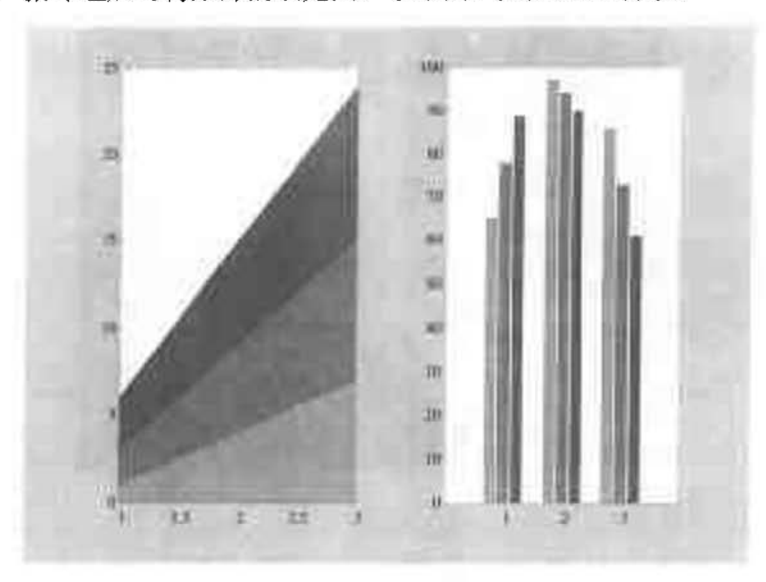


图 2-53　矩阵的填充图、条形图

【例 2-89】已知一学生高考成绩，试绘制成绩的平面饼图与三维饼图。

```
a=[85 68 95 74 65 49 ];
subplot (1,2,1) ,
pie (a,[1 0 0 1 0 0],{'数学','物理','化学','英语','语文','政治'}) ,
legend ({'1','2','3','4','5','6'}) subplot (1,2,2) ,
pie3 (a,a==min (a)) ,colormap (cool)
```

这组指令执行后可得成绩的平面饼图与三维饼图，如图 2-54 所示。

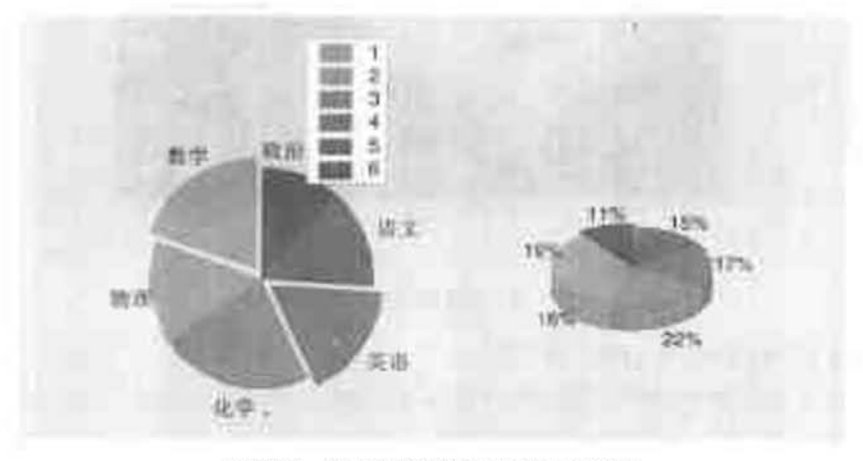


图 2-54　学生成绩平面饼图与三维饼图

【例 2-90】用二元函数画图指令在圆域上绘制 z=xy 的图形。

```
ezsurf ('x*y','circ') ;shading flat;view ([-18,28])
```

这组命令执行后可在圆域上绘制 z=xy 的图形，如图 2-55 所示。

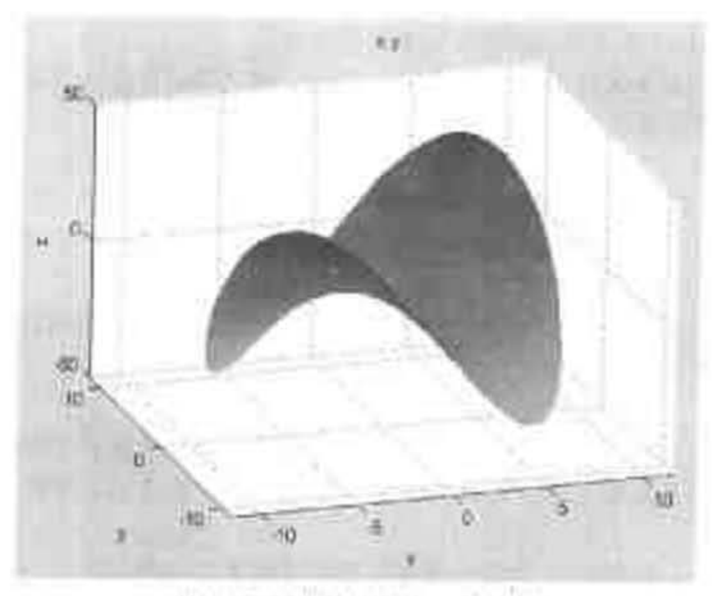


图 2-55 在圆域上绘制 z=xy 的图形

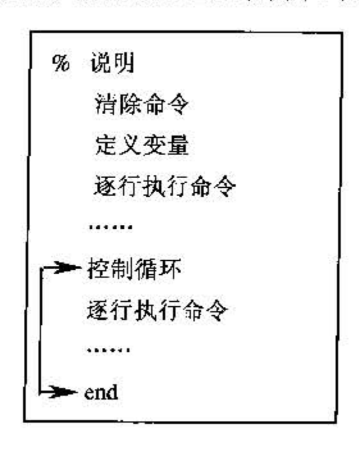
2.5 MATLAB 程序设计基础

MATLAB 是一种高效的编程语言。MATLAB 语言为解释性程序设计语言，对程序中的语句边解释边执行。MATLAB 程序与其他任何高级语言一样，是由顺序、选择和循环三种基本控制结构所构成。MATLAB 语句由表达式语句、赋值语句、控制语句、调试语句和空语句组成。控制语句包括条件语句、循环语句及一些转移语句。MATLAB 的任何一个语句键入后，键入 Enter 键（回车）即可执行，故又把语句称为命令。

2.5.1 MATLAB 程序设计基本规则

MATLAB 程序设计可以在 MATLAB 的编辑/调试器里进行，在编辑/调试器窗口能够简单方便、快捷高效地编辑与调试程序。

MATLAB 程序书写格式与函数的结构类似，也比较简单。程序的基本结构组成示意如下：



对于示意的程序基本组成结构作如下简要说明：

（1）工程问题 MATLAB 程序的分析、设计、调试和运行是一个完整设计过程，应充分利用编辑/调试器以及各种调试指令。

（2）% 后面的内容是程序的注释说明，包含有程序的名称、功能与对用户交代的问题等。它可以只有一行，也可有多行，它不是程序的正式内容，可以没有。

（3）在主程序开头用 clear 指令清除变量。注意在子程序中不要用 clear，以免将主程序传送到子程序的重要变量、参数清除了而丢失信息。

（4）已经介绍过，最好将程序文件放在 MATLAB 默认路径 MATLABR11/WORK 的目录下，这样可以省略程序的路径，否则 MATLAB 程序的路径与子目录必须填写完整。

（5）定义变量参数值要集中放在程序的开始部分，以便于以后的程序检查、维护。在语句行的最后输入分号时，执行语句行的结果不会显示在屏幕上，以提高执行速度。

（6）程序尽量模块化，也就是采用主程序调用子程序的方法。

（7）MATLAB 程序的语句与程序流程控制应符合 MATLAB 的语法规则。

（8）注意编辑/调试器用颜色区分程序内容的类别，分别为：

绿色 注释部分，程序并不执行；

黑色 程序主体部分；

红色 属性值设定或标识的部分；

蓝色 流程控制部分，比如 for，if…else 等语句；

以便一见字符颜色就会知道是哪一部分的内容。

2.5.2 表达式、表达式语句与赋值语句

1. 表达式

任何运算符按 MATLAB 的语法规则作用于操作对象（常量、变量、或函数）便构成表达式。在 MATLAB 的程序中有算术表达式、函数表达式、关系表达式、逻辑表达式等几种。MATLAB 中的数组可以进行这四种运算，而矩阵只能进行前两种运算。

2. 表达式语句

单个表达式就是表达式语句。一行可以只有一个语句，也可有多个语句。此时语句之间以分号“；”或逗号“，”或回车换行而结束。已经介绍过，MATLAB 语言中一个语句可以占多行，由多行构成一个语句时需使用续行符“…”；以分号结束的语句执行后不显示运行结果；以逗号或回车换行结束的语句执行后显示运行结果（即表达式的值）；表达式语句运行后，其表达式的值暂时保留在固定变量 ans 中。变量 ans 只保留最近的一次结果。

3. 赋值语句

将表达式的值赋值给变量构成赋值语句。

2.5.3 MATLAB 程序流程控制

MATLAB 程序设计也与其他高级计算机程序设计语言一样，有三大基本控制结构用

以实现各种不同的运算功能，即有顺序、选择、循环三个结构。那么程序流程的控制必定是这种程序设计语言的重要组成部分，并且有自身的语法规则。

2.5.3.1 分支选择语句

分支选择语句有两种：if条件分支语句与switch开关分支语句。

1. if条件分支语句

if-else-end 条件语句为程序流程控制提供了一种条件分支结构，这种语句有三种格式。这三种格式均以“if” 开头，以“end”结束。

（1）最简 if-end 语句格式为：

if 逻辑表达式

语句

end

这种条件语句执行规则是，计算 if 后面所跟逻辑表达式的值，当逻辑表达式的值为真（即 1）时，则执行 if 与 end 之间的语句，执行完之后继续向下执行；当逻辑表达式的值为假（即 0）时，将跳过“if”与“end”之间的语句而继续执行下面的语句。

【例 2-91】将一数组做特殊排列。

给出程序 L2501.m 如下。

```
% MATLAB PROGRAM L2501.m
%
for i=1:9;
  if i<=5
    a(i)=i;
  else
    a(i)=10-i;
  end
end
a
```

程序运行后特殊排列的数组为:

```
a =
     1     2     3     4     5     4     3     2     1
```

（2）if-else-end 条件语句格式为：

if 逻辑表达式

　语句 1

else

　语句 2

end

这种条件语句的执行规则是，当逻辑表达式的值为真时，则执行语句 1，然后跳过语句 2 向下执行；当逻辑表达式的值为假时，则执行语句 2，然后向下执行。

【例 2-92】将一数组做处理。

给出程序 L2502.m 如下。

```
% MATLAB PROGRAM L2502.m
%
a=[1 2 3;4 5 6];
i=2;j=3;
if i==j
   a (i,j) =8;
else
   a (i,j) =9;
end
a
```

运行这一段程序后得：

```
a =
    1     2     3
    4     5     9
```

（3）完整 if-elseif-end 条件语句格式为：

if 逻辑表达式 1
  执行语句 1
elseif 逻辑表达式 2
  执行语句 2
 ……
else
  执行语句 3
end

这种条件语句执行规则是，若逻辑表达式 1 的值为真时，则执行语句 1；如果为假，则要判断逻辑表达式 2 的值是真还是假，如果为真，则执行语句 2；否则逻辑表达式 2 的值也为假，则执行语句 3，然后向下执行。

【例 2-93】处理数组。

给出程序 L2503.m 如下:

```
% MATLAB PROGRAM L2503.m
%
a=[1 2 3;4 5 6;7 8 9];
i=2;j=3;
if i==j
   a (i,j) =0;
elseif abs (i-j) ==2
   a ((i-1) , (j-1)) =-1;
else
   a (i,j) =-5
```

```
end
a
```

运行这一段程序后得：

```
a =
     1     2     3
     4     5    -5
     7     8     9
```

2. switch 开关分支语句

switch 开关分支语句格式为：

switch 表达式（标量或字符串）

  case 值 1

      语句 1

  case 值 2

      语句 2

…………

otherwise

      语句 3

end

switch 开关分支语句执行规则是，判断表达式的值与哪种情况（case）的值相同，就执行哪种情况的语句；如果表达式的值与列出的所有情况（case）的值都不相同，则执行 otherwise 后的语句。switch 开关语句中可以不包括 otherwise，此时如果表达式的值与所有列出的情况（case）的值都不相同，则继续向下执行。

【例 2-94】绘制正弦曲线余弦曲线。

为解题给出程序 L2504.m 如下。

```
% MATLAB PROGRAM L2504.m
%
t=0:0.1:2*pi;
trigname=input ('Input trig function name:') ;
switch trigname
    case 'sin'
     plot (t,sin (t))
    case 'cos'
     plot (t,cos (t))
    otherwise
     break
end
```

运行这一程序后可以得 0～2π 范围的正弦曲线与余弦曲线，分别为图 2-56 与图 2-57 所示。

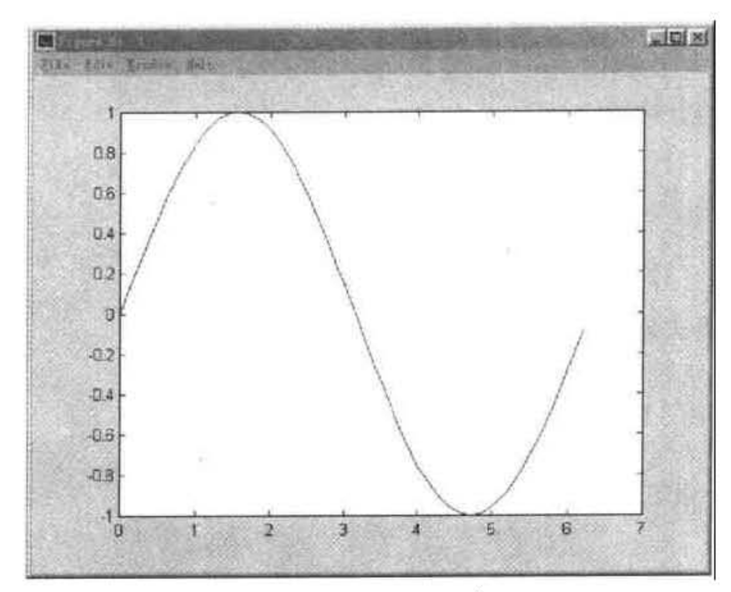


图 2-56 正弦函数曲线

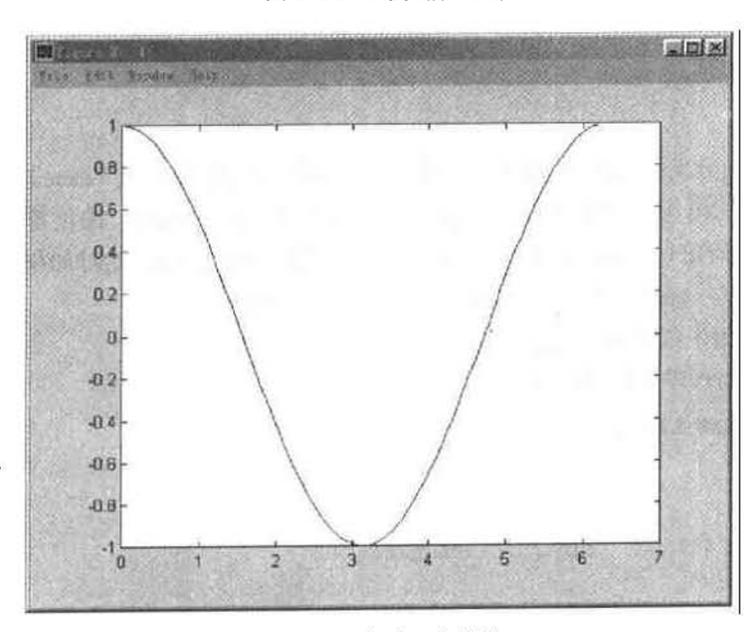


图 2-57 余弦函数曲线

2.5.3.2 *循环语句*

MATLAB 中的循环语句分为 for 循环语句（指定循环次数的）和 while 循环语句（不指定循环次数的）两种。

1．for 循环语句

for 循环语句的语句格式为：

for 循环变量=起始值：步长：终了值

循环体

end

for 型循环语句执行规则是：for 循环语句第一次循环先将循环变量的起始值赋值给循环变量，然后执行 for 与 end 之间的循环体，第二次循环则将循环变量的起始值与一个步长之和赋值给循环变量，再执行 for 与 end 之间的循环体，直到终了值为止，循环便终止，系统的控制转向循环语句之后的语句。

除此之外，还需要说明：for 循环语句的循环变量可以是一个数组；for 循环语句可以嵌套使用；for 循环语句的循环体内对循环变量的重新赋值不会终止循环的执行，也不会改变循环执行的结果；for 型循环语句可被等价的数组运算代替；为了提高运行效率，对循环体内所使用的数组应该预先分配其存储单元。

以下举例都是在 Word 里运行 Notebook 或在 MATLAB 命令窗口输入指令后并执行这些指令。

【例 2-95】计算控制系统 Bode 图横坐标 1～10 的对数刻度值。

为解题给出程序 L2505.m 如下。

```
% MATLAB PROGRAM L2505.m
%
for n=1:10;
   x(n)=log10(n);
end;
x
```

当程序运行后，显示结果如下：

```
x =
  Columns 1 through 7
   0    0.3010    0.4771    0.6021    0.6990    0.7782    0.8451
  Columns 8 through 10
   0.9031    0.9542    1.0000
```

【例 2-96】魔方阵行元素求和示例。

为解题给出程序 L2506.m 如下。

```
% MATLAB PROGRAM L2506.m
%
n=5;
x=magic(n)
for j=1:n
  s=0;
  for i=1:n
     s=s+x(j,i);
  end
s
ind
s1=(diag(x))'
```

```
sum (s1)
```

当程序运行后，显示结果如下：

```
x =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
s =
    65
s1 =
    17     5    13    21     9
ans=
    65
```

程序运行结果证明魔方阵行元素和与魔方阵的对角矩阵元素之和相等。

2. while 循环语句

while 循环语句的语句格式为：

while 表达式
　　循环体
end

while 循环语句执行的规则是：当表达式为真（即 1）时，则执行循环体，执行后再判断表达式的值是否为真，若表达式为真时，则继续执行循环体，直到判断表达式为假时为止，跳出循环体，继续往下执行。

由 while 循环语句执行的规则可见：一是 while 循环语句执行循环体的次数为不确定的，这是与 for 循环语句不同的；二是循环体的执行一定会改变 while 后面所跟表达式的逻辑值，否则这个 for 循环会是一个死循环。

【例 2-97】计算小于 100 的 Fibonnaci 数组。

为解题给出程序 L2507.m 如下。

```
% MATLAB PROGRAM L2507.m
%
f=[1,1]; j=1;
while f (j) +f (j+1) <100
f (j+2) =f (j) +f (j+1) ;
j=j+1;
end
f
```

当程序运行后，显示小于 100 的 Fibonnaci 数组如下：

```
f =
   1     1     2     3     5     8    13    21    34    55    89
```

2.5.4 MATLAB 程序流程控制指令

MATLAB 程序的语句流程除了有分支选择结构与循环结构来进行控制外，还有几个程序流程控制指令。

1. break （ ） 函数指令

break（ ）函数指令的作用是中断循环语句的执行。中断的循环语句可以是 for 语句，也可以是 while 语句。当满足在循环体内设置的条件时，可以通过使用的 break （ ） 函数指令使之强行退出循环，而不是达到循环终止条件时再退出循环。显然，循环体内设置的条件必须在 break （ ） 函数指令之前。对于嵌套的循环结构，break （ ） 函数指令只能退出包含它的最内层循环。

【例 2-98】计算介于$10^5 \sim 10^6$之间 n！数值的大小及其阶数。

为解题给出程序 L2508.m 如下。

```
% MATLAB PROGRAM L2508.m
%
n=1;
while prod(1:n)<1e100
   if prod(1:n)>1e5 & prod(1:n)<1e6
      break
   end
n=n+1;
end
prod(1:n)
n
```

程序运行后，求得介于$10^5 \sim 10^6$之间 n！数值的大小及其阶数为：

```
ans = 362880
n = 9
```

2. return （ ） 函数指令

return（ ） 函数指令的作用是中断函数的运行，返回到上级调用函数。return（ ）函数指令既可以用在循环体内，也可以用在非循环体内。

3. pause（ ） 函数指令

pause（ ） 函数指令是暂停指令。运行程序时，到 pause（ ） 函数指令执行后，程序将暂停，等待用户敲任一键后而继续执行。除此之外，pause（ ） 函数指令还有几种使用形式：形式 pause（n）的功能为暂停 n 秒；形式 pause off 的功能为其后的任何暂停语句无效；形式 pause on 的功能为其后的暂停命令有效。

2.5.5 MATLAB 程序的交互式指令

用户可以通过交互式指令协调 MATLAB 程序的运行。例如可以使用 MATLAB 的 input（）函数指令、yesinput（ ） 函数指令、keyboard（ ） 函数指令以及选择菜单功能等不同程度地响应程序运行过程中出现的各种提示。需要注意的是，由于交互式指令

无法在 Notebook 环境中运行，本节的例题都是在 MATLAB 命令窗口中运行的。

1. input（ ） 函数指令

input （ ） 函数指令是带有询问提示的输入指令。

input （ ） 函数指令格式 1 为：

input（string）

这种格式的功能是以文本字符串 string 为信息给出用户提示，并等待从键盘输入。键入的可以是当前工作空间中任意变量的 MATLAB 表达式（而不是字符串），其值作为 input（ ） 函数指令的返回值（同时可以赋值给一个变量）。

input （ ） 函数指令格式 2 为：

input（string,'s'）

这种格式的功能是以文本字符串 string 给出用户提示，并等待从键盘输入字符串（而不是表达式），键入的字符串作为 input（）的返回值（同时可以赋值给一个变量）。

2. yesinput（ ） 函数指令

yesinput （ ） 函数指令是带有询问提示和应答检查的智能输入指令。

yesinput （ ） 函数指令格式为：

answer=yesinput（question,default,possib）

这种格式的功能是执行该命令时，首先显示字符串 question 和缺省的应答值 default 作为输入提示。如果按下 Enter 键，则缺省值 default 就是默认的输入值。本命令自动检查应答的正确性。

命令中的输入参数：

question 询问字符串，显示在命令窗中作为对用户输入的提示。

default 指定数据的缺省值。本命令检查 default 是否为字符串，以断定读入变量的类型。

Possib 指定可供选择的应答范围：若期望应答的类型是字符串，则“possib”或者是一个字符数组，其每一行都可作为允许的应答；或者是一个由若干允许应答构成的字符串，各应答间以字符“|”（意为“或”）间隔。若期望应答的类型是数值，则“possib”必须是一个允许应答的上下限构成的 1x2 的向量：

possib（1）≤应答≤possib（2）

对于错误的应答，本命令会反复显示字符串 question 以提示用户重复输入。

命令中的输出参数：

answer 允许应答值。

3. keyboard（ ） 函数指令

keyboard 指令是调用键盘指令。

当 keyboard（ ） 函数指令用在一个 M-文件中时，执行该命令则程序暂停并将“控制权”转交给键盘。此时用户通过操作键盘可输入各种合法的MATLAB 指令。当用户使用 return（ ） 函数指令并敲 Enter 键后，“控制权”才交还给M文件。

4. menu（ ） 函数指令

menu （ ） 函数指令用来生成选择菜单。

menu （ ） 函数指令调用格式为：

k=menu（'menuname',s1,s2...sk）

menu （ ） 函数指令的功能是生成一个按钮式菜单系统，用于用户选择输入。其中字符串 si（i=1,..32）为菜单项，最多 32 项。k 为用户选择菜单号的返回值。

5．choices （ ） 函数指令

choices （ ） 函数指令是建立按钮式演示菜单指令。

choice 指令格式为：

choices（name,header,buttonlabels,callbacks）

choice （ ） 函数指令的功能是：开设一个窗口，由控制界面及其所对应的功能生成一个演示菜单。前两个参数是字符串（第一个是菜单窗口的名称，第二个是菜单窗口的标题），后两个是以字符串为行的文本矩阵。函数指令 choice（ ）以文本矩阵 buttonlabes 的每一行字符串为标签构造按钮菜单，从而生成一个用户界面。参数 callbacks 也是一个文本矩阵，其每行字符串都是用户设计的一个函数的名称。Buttonlabes 的每个按钮标签对应了 callbacks 中相应行的一行。用户按下 buttonlabes 中的一个按钮，choice 就可以调用由 callbacks 的相应行所给出的函数，从而实现一项功能。Buttonlabes 构成的控制界面是面向用户的，而 callbacks 的每项功能则是 MATLAB 的函数生成的命令。控制界面的最后一个附加按钮其标签为 Close。

2.5.6 MATLAB 程序的调试指令

MATLAB 程序设计完后，程序并不是也不可能完美无缺，没有任何问题。甚至有些设计的 MATLAB 程序根本不能运行。此时，一方面可以按程序的功能逐一检查其正确性；另一方面，可以用 MATLAB 程序的调试命令对程序进行调试。MATLAB 有多个调试函数指令，如 dbstop （ ）、dbup （ ）、dbdown （ ）、dbstep （ ）、dbquit （ ）等等。在 MATLAB 命令窗中，由执行 dbstop （ ）函数命令设置断点开始进入调试模式。此时用户可以跟踪程序（M-函数文件）的运行，考察程序执行状况，直到键入 dbquit （ ）函数命令而退出调试模式。

必须指出，调试命令不能用于非函数文件；Notebook 环境中无法使用调试指令；在调试模式下程序中断后命令窗中的提示符为 K。

1．dbstop （ ）函数指令

dbstop （ ）函数指令的功能是设置断点。用来临时中断一个函数文件的执行，给用户提供一个考察函数局部变量的机会。dbstop （ ）函数指令格式有以下 4 种使用形式：

形式 1　**dbstop　at** <行号> **in** <文件名>

这种格式的功能是：在指定文件的指定行处设置断点。

形式 2　**dbstop in** <文件名>

这种格式的功能是：在函数文件的第一个可执行处设置断点。

形式 3　**dbstop if erro**

这种格式的功能是：在任何函数文件运行出错时产生中断（不包括语法错误）。出错中断后用户不能恢复程序运行。

形式 4 **dbstop if naninf** 或 **dbstop if infnan**

这种格式的功能是：当出现 INF 或 NaN 值时产生中断。

2. dbclear（ ）函数指令

dbclear（ ）函数指令用来清除断点。

dbclear（ ）函数指令的功能是用来清除由相应 dbstop（ ）函数命令设置的断点。dbclear（ ）函数指令有以下 6 种使用格式：

形式 1 **dbclear at <行号> in <文件名>**

这种格式的功能是：清除指定文件在指定行处的断点。

形式 2 **dbclear all in**〈文件名〉

这种格式的功能是：清除指定文件中所有断点。

形式 3 **dbclear in**〈文件名〉

这种格式的功能是：清除第一个可执行处的断点

形式 4 **dbclear all**

这种格式的功能是：清除所有文件的所有断点。

形式 5 **dbclear if error**

这种格式的功能是：清除由 dbstop if error 语句设置的断点。若此命令之后出现错误，MATLAB 将中断当前运算而返回基本工作空间。

形式 6 **dbclear if naninf** 或 **dbclear if infnan**

这种格式的功能是：清除 dbstop 关于 INF 或 NAN 的断点。

3. dbcont（ ）函数指令

dbcont（ ）函数指令用来恢复对于执行 dbstop（ ）函数指令或 dbsetup（ ）函数指令而导致中断（中断后的提示符为 k）的程序。用 dbcont（ ）函数指令恢复程序执行，一直到遇到其他已设置的断点或出现错误，或返回基本工作空间。

4. dbdown（ ）函数指令

dbdown（ ）函数指令用来改变工作空间到下级函数。在调试模式下，dbdown（ ）函数指令用来改变 0 与 dbup（ ）函数指令相关联的局部工作空间。dbdown（ ）函数指令改变当前函数的工作空间到其所调用的下级子函数，即跟踪到下级子函数。

5. dbup（ ）函数指令

dbup（ ）函数指令用于改变工作空间到上级函数。在调试模式下，dbup（ ）函数指令改变当前函数的工作空间到调用该函数的上级函数（或基本工作空间）。

6. dbstep（ ）函数指令

dbstep（ ）函数指令用于执行一行或多行代码。在调试模式下，dbstep（ ）函数指令允许用户实现逐行跟踪。

7. dbstack（ ）函数指令

dbstack（ ）函数指令用于列出调用关系。在调试模式下，dbstack（ ）函数指令显示当前的文件名及中断点的行号，其调用者的、调用者的调用者的…文件名和断点行号，直到最外层的函数文件。

8. dbstatus（ ）函数指令

dbstatus（ ）函数指令用来列出全部断点。在调试模式下，执行以下格式的指令：

dbstatus（文件名）

可列出指定文件的全部断点。

9．dbtype（）函数指令

dbtype（）函数指令用来显示带行号的文件内容，以协助用户设置断点。该命令有两种格式：

形式 1 **dbtype**（文件名）

这种格式的功能是：列出指定文件的全部内容，每行前带有行号。

形式 2 **dbtype** <文件名><始行号：><末行号>

这种格式的功能是：列出指定文件自起始行号到末行号部分的内容。

10．dbquit（）函数指令

dbquit（）指令用来退出调试模式。

在调试模式下，dbquit（）函数指令立即强制终止调试模式，将控制转向基本工作空间。此时，函数文件的执行没有完成，也没有产生返回结果。

2.5.7 MATLAB 程序的常用警示指令

为了提请程序使用者应该注意与防止出现的问题，以及关注程序运行、判断出错原因并指挥程序继续运行与否等等，在设计 MATLAB 程序时，常常使用一些警示函数指令。常用的警示函数指令及其功能如下：

error（'message'） 显示出错信息 message，终止程序运行。
errortrap 程序出错后是否继续执行的双位开关。
Lasterr 显示 MATLAB 自动判断的最新出错原因，并终止程序运行。
warning（'message'） 显示警告信息 message，程序继续运行。
lastwarn 显示 MATLAB 自动给出的最新警告提示，程序继续运行。

2.6 MATLAB 文字处理工具 Notebook

MATLAB Notebook 就像一个会进行运算的文稿本，在该文稿本中，可以编辑文字，可随时修改计算命令、随时计算并绘制图形。这对于撰写科技报告、论文、专著的科学工作者与教师，以及对于演算理工科习题的广大学生、研究生，MATLAB Notebook 都是非常有用的，这正是本书要介绍 Notebook 工具的原因。

本书正文的编辑以及其中 MATLAB 程序的运行，除开带鼠标操作的图形交互指令以外都是在 Notebook（使用 M-book 模板）集成环境下完成的。

2.6.1 Notebook 简介

众所周知，当今世界微软公司的 Word 软件在字处理方面功能最强，而 MATLAB 的数值计算功能最优，如果能够将两者结合起来，就能集二者之所长。

MATLAB Notebook 就是一个兼有两者优点的工具。这个工具是 MathWorks 公司为文字处理、科学计算、工程设计而开发的一个集成运行环境，它不仅有 Word 的全部功能，而且还具备 MATLAB 无与伦比的数学解算功能和灵活自如的计算结果可视化能力。

在 MATLAB Notebook 中有一个称为 M-book 的特殊文档，就是 Word 中使用的模板文

件 M-book.dot。所谓启动 Notebook，就是指从不带【Notebook】菜单项的 Word 常规窗口变成已启用 M-book 模板的 Word 窗口，此时 Word 窗口主菜单已增添了【Notebook】菜单项。

一个 M-book 被打开时，MATLAB 即随之启动。当创建和编辑一个 M-book 时，Word 将启动 M-book 模板。正是由于 M-book 模板,才可以在 Word 和 MATLAB 两者之间交互。一个 M-book 文档就是一个交互式 MATLAB 过程的记录。在这个记录中，包含了文字、MATLAB 命令、计算结果和图形等，并可以借助 Word 的功能对之进行编辑处理。

Notebook 是这样工作的：用户在 Word 文档中输入的命令送到后台的 MATLAB 中运行，然后 MATLAB 将计算结果和绘制的图形送回到 Word，并插入到文档中。

MATLAB Notebook 所需要的系统环境已经在 2.1.1 节中介绍过，这里不再重复。

2.6.2 创建与打开 M-book 文档

创建与打开 M-book 文档就是启动 Notebook，即在 Word 中编辑的文件使之具有 MATLAB 运算功能。创建与打开 M-book 文档有以下几种方法。

（1）使用 Notebook 之前，必须先启动 Word。在其主菜单的【文件】菜单项的下拉菜单中选择【新建】子项，即弹出图 2-58 所示的对话框。在图 2-58 中选中“M-book.dot”模板图标，并选中图 2-58 右下角的“文档”单选钮，按下【确定】按钮，就会创建一个新的 M-book 文档，并在文档的菜单栏上添加【Notebook】菜单项（图 2-59）。

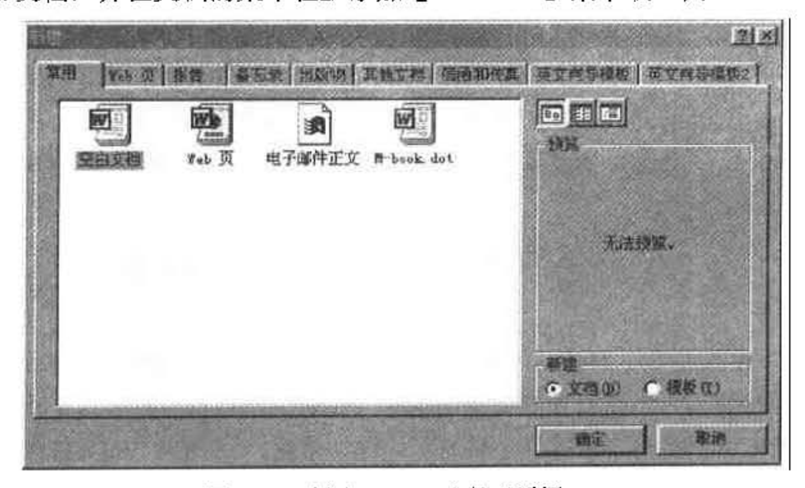


图 2-58　创建 M-book 文档对话框

（2）如果安装 MATLAB5.3，则可在 MATLABR11\Notebook\pc 的路径上找到 M-book.dot 模板；如果安装 MATLAB5.1 或 5.2，可在 Microsoft Office\Office 的路径上找到 M-book.dot 模板。不论通过“我的电脑”或者“程序管理器”或者“文件管理器”，只要进入“浏览器”中，双击 M-book.dot 图标，即可创建一个新的 M-book 文档（图 2-11）。

（3）在 Microsoft Windows 平台上，可在 MATLAB 的命令窗口中键入下面的命令来新建一个 M-book 文档：

notebook

或在 MATLAB 的命令窗口中键入下面的命令来打开已建的 M-book 文档：

notebook “已建的文件名”

需注意，第一，notebook 后要空格；第二，已建的文件名前要带文件的路径，只有文件存放在 MATLABR11/WORK 路径下时才可省略。

（4）如果要将其他的文档（例如 .wps 或.dot 文件）转换为 M-book,可采用下面的步骤：

①新建一个 M-book。

②选中菜单栏的“插入”菜单中的文件选项。

③选中想转换的文件。

④单击“确定”按钮。

2.6.3 Notebook 菜单命令简介

打开或新建一个 M-book 时，Word 的主菜单栏中将增加一个【Notebook】菜单项，如图 2-59 所示。Notebook 下拉子菜单共有 15 个菜单命令，Notebook 的所有功能都是通过该子菜单的选项命令实现的。表 2-15 列出了所有这些选项及其功能。

表 2-15 Notebook 菜单项及取功能

选　　项	作　　用
Define Iput Cell	定义输入细胞
Define AutoInit Cell	定义自活细胞
Define Calc Zone	定义计算区
Undefine Cells	将细胞转换为文本
Purge Output Cells	从所选篇幅中删除所有输出细胞
Group Cells	生成细胞群
Ungroup Cells	将细胞群转换为输入细胞或自活细胞
Hide （Show ）Cell Markers	是否显示生成细胞的中括号
Toggle Graph Output for Cell	是否嵌入生成的图形
Evaluate Cell	运行细胞
Evaluate Calc Zone	运行计算区
Evaluate M-book	运行整个 M-book 中的所有输入细胞
Evaluate Loop	多次运行输入细胞
Bring MATLAB to Front	把 MATLAB 命令窗口调到前台
Notebook Options	设置数值和图形的输入格式

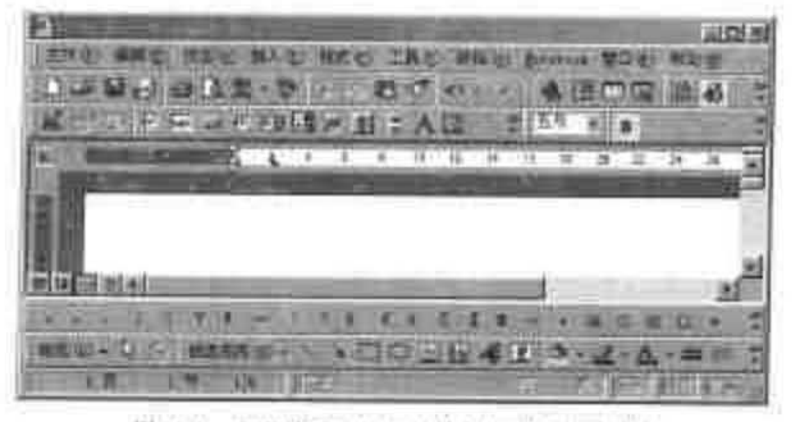


图 2-59 打开或新建 M-book 的 Word 的主菜单栏

2.6.4 Notebook 的使用

Notebook 某种意义上即是 Word 中的 M-book。Notebook 是通过动态链接来和 MATLAB 交互的。Word 中的 M-book 和 MATLAB 交互的基本单位为细胞。在 Notebook 中，Word 与 MATLAB 之间交换的信息，叫做"细胞（群）"。M-book 需要把在 Word 中输入的 MATLAB 命令或者语句行组成细胞（群），传送到 MATLAB 中运行，运行输出的结果再以细胞（群）的方式传回 M-book 中。

2.6.4.1 输入、输出细胞与使用

所谓"输入细胞"，是由 M-book 送向 MATLAB 中运行的指令。输入细胞可包含一行，也可包含多行；可以是只包含嵌在文本之中的命令，还可以是 MATLAB 程序。定义细胞的目的是要使用它，即在 Word 中运行 MATLAB，使之返回运算结果。由 MATLAB 返回到 M-book 的计算结果叫做"输出细胞"。

1. 生成输入细胞

生成输入细胞的具体步骤如下：

首先，在 Word 中，用文本格式输入指令，在命令结束时不要按回车键或空格键。然后，在【Notebook】菜单中选中【Define Input Cell】命令项或者按下组合键【Alt+D】。

完成以上操作后，Notebook 即用普通文本输入的命令生成输入细胞，并用黑色中括号把指令括起来，括号中的指令为绿色字体显示，看起来很醒目。

2. 生成输入细胞与把命令送到 MATLAB 中运行

生成输入细胞与把命令送到 MATLAB 中运行合在一起，其步骤为：

首先，采用文本格式输入命令，在命令结束时不要按回车键或空格键，让光标停在命令的末尾。其次，在【Notebook】菜单中选中【Evaluate Cell】选项或者按组合键【Ctrl+Enter】。完成以上操作后，生成的输入细胞，是用黑色中括号括起来的，括号中的指令显示为绿色字体，并在其后跟着显示蓝色字符的运算结果。

需要说明的是，可以用鼠标的操作代替选中并执行【Define Input Cell】项命令的操作，即用鼠标将 Word 中的文本格式指令选中，有的文献上叫做"点亮"或"刷黑"。

当输入的命令为多段或比较大时，应选中所有的命令行来定义输入细胞或者直接运行。如果仅在输入结束后而不选中所有的命令行就运行命令或定义输入细胞，则只有最后一行被执行或定义。

Notebook 采用了丰富多彩的颜色分别显示不同的操作内容：输入细胞用黑色方括号把指令括起来，括号中的指令为绿色字体显示；输出细胞输出 MATLAB 的运算结果时，也用黑色方括号把结果括起来；输出结果放在输入细胞的后面。Notebook 采用蓝色字符显示计算数值结果和文本结果；采用红色字符显示错误信息。需要说明的是，这些色彩在计算机显示器上一目了然，而在书本上是不能看到的。

2.6.4.2 自活细胞与使用

1. 自活细胞的概念

用【Notebook】菜单中【Define AutoInit Cell】命令定义的输入细胞，叫做自活细胞。

采用自活细胞的方式来定义 MATLAB 命令的目的是，无须任何别的操作就可以很方便地恢复上次的 MATLAB 工作空间。

自活细胞和输入细胞的不同之处在于：其一，每次打开 M-book 时会自动运行自活细胞，而不能自动运行输入细胞；其二，自活细胞字符用深蓝色来显示，而输入细胞用绿色显示其字符，但两者均用黑色方括号括起来。

自活细胞的来源有两种，不论对用文本格式输入 MATLAB 命令还是对输入细胞，选中后均可用【Notebook】菜单中的【Define AutoInit Cell】项命令生成。对用输入细胞生成方式的操作，可以选中或把光标放在输入细胞的末尾，选择【Notebook】菜单中的【Define AutoInit Cell】项命令把输入细胞转换为自活细胞。定义自活细胞也就是创建自活细胞。

2．自活细胞的使用

自活细胞的使用就是运行自活细胞，即在 Word 中运行 MATLAB，使之返回运算结果。运行自活细胞的步骤为：

首先，把光标方在自活细胞中，然后，选择【Notebook】菜单中的【Evaluate Cell】项或按按组合键【Ctrl+Enter】。

2.6.4.3 细胞群及其使用

1．细胞群的概念

在 Notebook 中，把包含多句 MATLAB 命令的多行输入细胞或自活细胞称为细胞群。所以，Notebook 允许多行 MATLAB 命令当作一个细胞群整体来运行。MATLAB 程序一般都有多行语句，所以应该把一个在 Word 中的 MATLAB 程序当作一个细胞群。

对于当作一个整体的细胞群，Notebook 中有一些运行规则：在细胞群中，不能包含 Word 文本和输出细胞；细胞群的输出结果用一个输出细胞显示并紧跟在细胞群的后面；当生成细胞群时，除非第一行为自活细胞，否则，Notebook 将把该细胞群当作输入细胞；如果细胞群中的一条命令会生成文本或数值形式的结果，而另一句会生成图形结果，则文本或数值结果总放在图形的前面。

2．细胞群的创建

当用多条命令生成一个图形时，细胞群是必需的。采用细胞群把这些命令生成一个输入细胞，最后绘制的结果将为一个输出细胞；如果把每一行命令都生成一个输入细胞，则输出的结果为多个输出细胞。

创建或生成细胞群的具体步骤为：

首先，选中将要生成细胞群的所有输入细胞；然后，选中【Notebook】菜单中的【Group Cells】选项命令执行或者按组合键【Alt+G】。必须注意，如果选中内容中包含输出细胞，Notebook 将自动删除输出细胞；如果选中内容中包含普通的 Word 文本，第一个输入细胞前的文本位置不变，第一个输入细胞后的文本全部放在细胞群后；当选中输出细胞的一部分而没有选中该输出细胞的输入细胞，Notebook 将自动把输入细胞选入细胞群。

若是 Notebook 把选中的多行输入细胞或自活细胞已转换为细胞群，则会去掉原来多个细胞的黑色方括号，用一对方括号把所有内容括起来。

3. 细胞群的运行

细胞群的运行与输入细胞相同：将光标放在细胞群中的任何位置，选中【Evaluate Cell】选项命令后执行或按组合键【Ctrl+Enter】。运行细胞群后，输出结果用一个输出细胞显示，默认情况下，Notebook 把输出细胞紧放在细胞群的后面，当细胞群已有一个输出细胞时，无论输出细胞位于 M-book 的任何位置，输出结果都将在该输出细胞中显示。

2.6.4.4 计算区及其使用

1. 计算区的概念

计算区（Calc Zone）是 M-book 中连续的一块，它把 M-book 分为几个相互独立的部分，这些部分包括：描述一个特定问题或特定操作的文本、输入细胞、输出细胞。当定义一个计算区时，Notebook 将该部分和 M-book 中的其他部分独立出来。

一个 M-book 文件可包含任意个计算区。计算区对于较为独立的问题很有用。值得注意的是，计算区中的变量并不仅仅在该计算区中起作用。

2. 计算区的生成

计算区的生成的步骤为：首先，用光标选中所有欲包含在一个计算区中的文本、输入细胞和输出细胞。当选取输入细胞和输出细胞时，必须选中输入细胞和输出细胞的所有内容。然后，选中【Notebook】菜单中的【Define Calc Zone】选项命令并执行或者按组合键【Alt+Z】。

3. 计算区的运行

计算区的运行与细胞群的运行相同：首先，把光标置于计算区内的任何位置；然后，选中【Notebook】菜单中的【Evaluate Calc Zone】选项命令执行或者按组合键【Alt+Enter】。

须注意，Notebook 默认情况下，计算区的输出细胞放在第一个输入细胞后面；当计算区已有一个输出细胞时，无论输出细胞在什么地方，输出结果将在输出细胞中显示。

2.6.4.5 细胞循环运行及其使用

Notebook 提供循环运行细胞的命令。循环运行细胞的具体步骤如下：

首先选中欲重复运行的输入细胞（一定要是绿色字符的输入细胞）；接着在【Notebook】菜单中选中【Evaluate Loop】选项命令或者按组合键【Alt+L】，将弹出如图 2-60 所示的对话框。在图 2-60 中选择各个选项进行设置：在“Stop After”区域中输入重复运行次数，单击【Stat】按钮开始运行 Notebook，同时【Stat】按钮变为【Stop】，

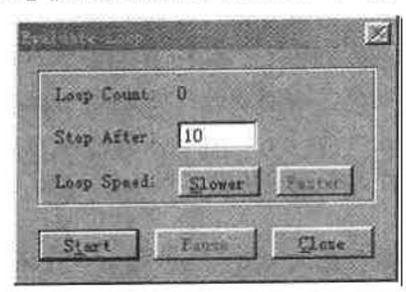


图 2-60 循环运行对话框

并显示已运行的次数；在"Loop Speed"区域中单击【Slower】或【Faster】按钮，可以改变每一次循环的快慢速度。点击【Pause】或【Continue】按钮，循环可暂停或继续运行；点击【Stop】按钮循环停止运行。

2.6.4.6 输出细胞的控制

MATLAB 以输出细胞将计算数值结果、图形、提示的错误信息返回到 M-book 中。M-book 模板中缺省设置的细胞字体样式如表 2-16 所示。

表 2-16 M-book 模缺省设置细胞样式

对象名称	字　体	字体大小	是否黑体	颜　色
一般文本	Times New Roman	10		黑色
输入细胞	Courier New	10	Bold	深绿
输出细胞	Courier New	10		蓝色
自活细胞	Courier New	10	Bold	深蓝
错误信息	Courier New	10	Bold	红色

输出细胞其他格式的控制，可以通过【Notebook】菜单中的【Notebook Options】选项命令执行后的对话框（图 2-61）来进行设置。



图 2-61 Notebook 设置对话框

（1）在"Numeric Format"区域中，可以设置的输出数据格式有 8 种："Short"、"Long"、"Hex"、"Bank"、"Plus"、"Short e"、"Long e"、"Rationsl"等。

（2）"Loose"与"Compact"选项，可以设置在输入细胞与输出细胞之间是否加空白行。选"Loose"时加空行，选"Compact"时不加空行。设置后须按下按钮【OK】才生效。

（3）在"Figure Options"区域中，选项"Embed Figures in M-book"被选中是默认

值，所有的图形输出都以输出细胞的方式嵌在M-book文档中；不选中该选项，图形将以独立的图形窗口输出，与单独使用MATLAB一样。

（4）在“Figure Options”区域中，选项“Use 16-Color Figures ”被选中是默认值，所有的图形输出都以16种颜色打印；不选中该选项，图形将以256种颜色打印。

（5）选项“Height ”与“Width”用来控制嵌入到M-book中图形的尺寸大小，“Height ”为图形的高，“Width”为图形的宽。其“Units”选项是设置尺寸大小的单位：英吋或厘米等。

2.6.4.7 Notebook使用的几个问题

（1）M-book文档中的MATLAB指令与标点符号都必须在英语状态下输入。

（2）带鼠标操作的图形交互指令不能在M-book文档中运行。

（3）MATLAB指令在M-book文档中运行的速度比在MATLAB命令窗口中慢很多。

（4）可将细胞（群）转换为普通文本。细胞（群）在Word里编辑的文件中是绿色的字符，这会影响Word文件的印刷效果。可以将Notebook中的输入细胞、自活细胞、细胞群转换为普通文本，其转换步骤为：

先将光标放在细胞中；然后选中【Notebook】菜单中的【Undefine Cells】选项命令或者按组合键【Alt+U】。

当细胞（群）被转换为文本之后，细胞（群）中的内容就用一般的文本格式显示，黑色的方括号对也被取消。当把输入细胞转换为文本时，该输入细胞的输出细胞自动转换为文本，所有绿色的字符均变成黑色字符。

（5）使用【Notebook】菜单中的【Bring MATLAB to Front】命令或者按组合键【Alt+M】可把MATLAB的命令窗口调到前台。

（6）使用【Notebook】菜单中的【Toggle Graph Output for Cell】命令可以控制是否显示输入细胞或输出细胞的输出图形。若控制为不输出图形，可将光标置于欲运行的细胞（群）内，选中该【Toggle Graph Output for Cell】命令项，在细胞（群）后将生成“no graph”，运行细胞（群）就不输出图形；再一次选中该项，即可输出图形。

第3章　MATLAB仿真集成环境工具SIMULINK

本书介绍的对象系统是MATLAB 5.3版，本章所研究的主要仿真工具理所当然的是 MATLAB 5.3 所采用的 SIMULINK 3.0 系统。SIMULINK（Dynamic System Simulation）是MATLAB里的一个实现动态系统建模、仿真与分析的软件工具包。

MATLAB中的TOOLBOX 工具箱与 SIMULINK 仿真工具，是控制系统计算与仿真最先进的工具。所以本章要对SIMULINK仿真工具作全面的介绍与深入的讨论。全章共9节。3.1节对 SIMULINK 仿真工具作简要介绍；3.2节介绍 SIMULINK的界面与菜单；3.3节介绍用SIMULINK建立系统模型；3.4节介绍SIMULINK仿真算法的数学基础；3.5节介绍SIMULINK仿真的数值分析；3.6节介绍观察SIMULINK的仿真结果；3.7节介绍SIMULINK的分析工具；3.8节介绍SIMULINK 3.0的自定义模块库与自定义子系统；3.9节介绍SIMULINK的Mask功能。

3.1　SIMULINK仿真工具简介

3.1.1　SIMULINK仿真工具概述

1. MATLAB及其SIMULINK

Mathworks开发的SIMULINK是MATLAB里的工具箱之一，主要功能是实现动态系统建模、仿真与分析，从而可以在实际系统制作出来之前，预先对系统进行仿真与分析，并可以对系统做适当的实时修正或者按照仿真的最佳效果来调试及整定控制系统的参数，以提高系统的性能，减少设计系统过程中反复修改的时间，实现高效率地开发系统的目标。

Mathworks从MATLAB 4.0版开始应用SIMULINK，当时它放在MATLAB 4.0版里的核心执行文件中。在MATLAB 4.2及以后的版本中，SIMULINK则以MATLAB里的工具包形式单独出现，即需单独安装。而在MATLAB 5.0版中，SIMULINK已升级为2.0版，在MATLAB 5.3 版中，SIMULINK则升级为3.0版。SIMULINK 3.0版比从前的版本有很大的改进。

需要特别说明的是，在MATLAB 5.3版的SIMULINK 3.0版中，其文件类型为.mdl，这与MATLAB 4.x不同，只要把老版本SIMULINK里设计好的.m文件改变其文件后缀名为.mdl即可在MATLAB 5.3版的SIMULINK 3.0环境下运行。

2. SIMULINK 的适用对象

SIMULINK 支持连续与离散系统以及连续离散混合系统，也支持线性与非线性系统，还支持具有多种采样频率（Multirate）的系统，也就是不同的系统能够以不同的采样频率进行组合，以仿真较大较复杂的系统。

3. SIMULINK 的模块库与模型窗口

在 MATLAB 的命令窗口里输入“SIMULINK”命令然后按回车键；或者连续选择 MATLAB 命令窗口中的【File】下的【New】下的【Model】选项命令或者选择【 】按钮选项均可进入 SIMULINK 浏览器即仿真器或模块库（图 3-1）。再点击图 3-1 上部工具栏的【 】按钮，就可进入“untitled”空白设计区桌面即模型窗口（图 3-2）。

与以往的版本相比，MATLAB5.3 的 SIMULINK3.0 有了很大的改变，新版把一些控制工具箱例如控制系统工具箱（Control System Toolbnox）、模糊逻辑工具箱（Fuzzy Logic Toolbox）、非线性控制设计模块（NCD Blockset）等等已编入 SIMULINK 里。打开 SIMULINK 浏览器（图 3-1），可以看到两层目录（图 3-2）结构。

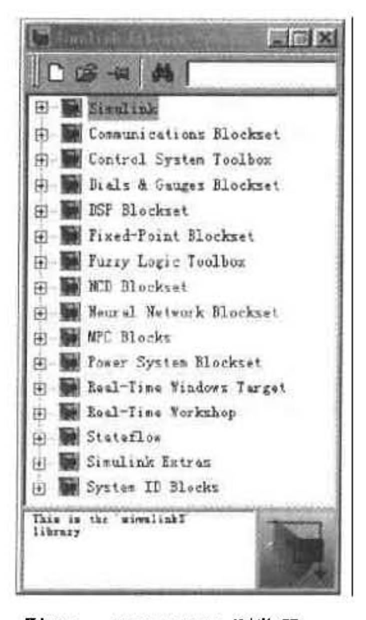



图 3-1　SIMULINK 浏览器

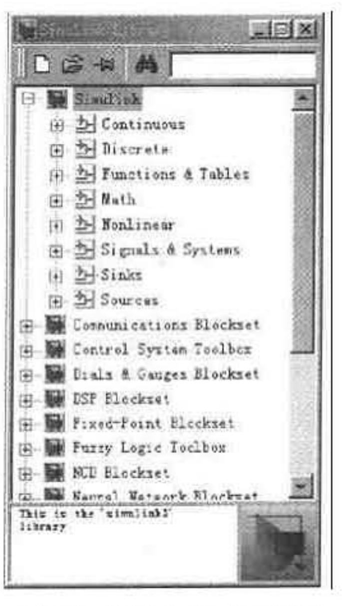


图 3-2　SIMULINK 两层目录

由图 3-1 可见，与 SIMULINK 并列的还有 14 项作为 SIMULINK 浏览器中的第一层目录。选中第 1 项 SIMULINK 后，下级有 8 项子目录（图 3-2），即是 SIMULINK 3.0 提供的 8 类基本模块：计有 Continuous（连续系统模块）、Discrete（离散系统模块）、Function & Tables （函数与表模块）、Math（数学运算模块）、Nonlinear （非线性系统模块）、Signals & System （信号与系统）、Sinks（输出模块）、Sources（输入源模块）等标准模块库。每个标准模块库中存储有多个相应的基本功能模块，供用户选择使用。

用户也可以创建与定制自己的功能模块，而不一定只使用 SIMULINK 系统软件提供的标准模块。这样，用户就可以自行扩充软件的应用范围。

SIMULINK 模型窗是 MATLAB 窗口系列中的一种特殊类型，在 SIMULINK 3.0 的窗口里有 File、Edit、View、Simulation、Format 与 Tools6 个菜单选项。

4. SIMULINK 的建模

SIMULINK 为用户提供了用方框图进行系统建模的图形窗口，采用这种建模方式来绘制控制系统的动态模型结构图，就像用笔与纸来画图一样容易，且更简单、准确而快捷。它与传统的仿真软件采用微分方程或者差分方程建模相比，具有更直观、更简单方便与更灵活的优点。这样可大大简化设计流程，减轻设计负担，降低设计成本，提高工作效率。

根据实际工程中控制系统的具体构成，将上述模块库中提供的各种标准环节拷贝到 SIMULINK 的模型窗口中，再用 SIMULINK 的连线方法连接成一个完整的 SIMULINK 动态结构图。各个环节可按 SIMULINK 特定的方法改变或设定其参数以与实际控制系统相对应。

在对于较大的系统建立模型时，SIMULINK 提供了系统分层排列的功能。SIMULINK 可将系统分为从高级到低级的好几层，每层又可以分为好几个小部分；每层系统模型创建完成后，再将其连接起来就是一个完整的系统了。分层的功能使我们可以方便地组织系统。

这与以前仿真系统大不一样。以前仿真控制系统需要将系统模块函数转化为微分方程或差分方程，再用某种程序设计语言（如 C 或是 FORTRAN 等）去程序化系统。但是对于 SIMULINK 的用户而言，只需要用鼠标的点击-拖拽功能，将模块库中提供的各种标准模块拷贝到 SIMULINK 的模型窗口中，就可以轻而易举地完成模型的创建，这样就大大降低了仿真的难度，至少用户不用为了仿真控制系统去专门学习某种程序设计语言。

5. SIMULINK 的仿真

在定义完一个模型以后，可以通过 SIMULINK 的菜单或者 MATLAB 命令窗口键入命令来对系统进行仿真以及分析其动态特性。对于使用者而言，菜单方式交互性强，非常方便；而命令方式对于运行某些方面的仿真程序是非常有用的。SIMULINK 内置的分析工具包括各种仿真算法、系统线性化、寻找平衡点等等。采用 Scop 模块与其他的画图模块，可以在仿真进行的同时，就观看到仿真结果。除此之外，使用者还可以在改变系统参数后观看系统中发生的迅速变化情况。仿真的结果还可以变量的形式保存起来，并将其输入到 MATLAB 的工作空间里做进一步的分析、处理或利用。

3.1.2 SIMULINK for Windows 仿真工具的安装

1. SIMULINK 对系统的要求

MATLAB 是在 Windows 下运行的程序设计语言，而 SIMULINK For Windows 是 MATLAB 里的专用软件，所以 SIMULINK 对系统的要求依赖于 Windows 及 MATLAB 的要求，它所需要的系统硬件配置与软件环境不必专门增加或改变，只要能满足 Windows 系统及 MATLAB 的要求即可。

2. SIMULINK For Windows 的安装

SIMULINK 3.0 的安装是与 MATLAB 5.3 的安装一起进行的，即通过光盘安装。安装 MATLAB 时（参见图 2-3），在打开的 MATLAB 安装内容选择对话框中，只要用鼠标点击即“勾选”SIMULINK、SIMULINK help（PDF）、SIMULINK help（HTML）选项，MATLAB 5.3 安装结束时，SIMULINK 3.0 也随之安装完成。

3.1.3 SIMULINK 仿真工具子目录及文件

MATLAB 5.3 及其 SIMULINK 3.0 安装完毕后，在 MATLABR11\TOOLBOX 子目录中，SIMULINK 系统子目录及文件有：

（1）MATLABR11\TOOLBOX\SIMULINK\BLOCKS

这个仿真工具块里有 85 个文件，占 2 331 503 B。

（2）MATLABR11\TOOLBOX\SIMULINK\DEE

这个仿真工具块里有 27 个文件，占 138 773 B。

（3）MATLABR11\TOOLBOX\SIMULINK\SIMDEMOS

这个用于演示的仿真工具块里有 96 个文件，占 1 330 913 B。

（4）MATLABR11\TOOLBOX\SIMULINK\SIMULINK

这个仿真工具子目录里有 111 个文件，占 2 815 239 B。

这 4 个子目录里绝大部分文件是 M 文件。

3.2 SIMULINK3.0 的界面与菜单

一个系统软件的界面形象往往会给人们留下深刻的印象，一个软件的学习也往往会从对这个软件界面各个组成部分逐个元素的了解而开始。特别是 Windows 下的软件界面，色彩艳丽、内容丰富、形象生动，也会给学习增添不少的新颖感觉与乐趣。

3.2.1 SIMULINK 3.0 的界面形象

MATLAB5.3 的 SIMULINK3.0 模块库浏览器的界面已经见过，在此仅将“Simulink”目录及其下属 8 个子目录的界面再次给出，如图 3-3 所示。在界面最上方标题栏“Simulink Library Browser”的右侧是最小化、最大化、关闭按钮，标题栏下是工具栏（左边）与待查关键词填写栏（右边），中间部分是模块库与模块两层目录的树状结构一览表，最下方是所选模块对象（Simulink）的文字说明“This is the 'simulinks' library”与图形标识，这些都是与过去的 SIMULINK 不一样的。

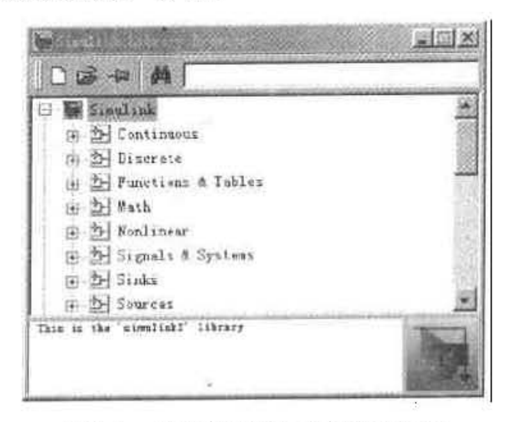


图 3-3 SIMULINK3.0 模块库浏览器

工具栏里有四个按钮：、是新建与打开模型文件按钮，与 Windows 系统的工具相同；是将模块库浏览器放在桌面最上层的按钮；是查找模块库中的模块按钮。

3.2.2 SIMULINK 功能模块的分类及其用途

SIMULINK 3.0 提供的 8 类基本模块库：Continuous（连续系统模块库）、Discrete（离散系统模块库）、Function & Tables（函数与表模块库）、Math（数学运算模块库）、Nonlinear （非线性系统模块库）、Signals & System （信号与系统模块库）、Sinks（输出模块库）、Sources（输入源模块库）等标准模块库。用鼠标左键选中图 3-3 中 Simulink 下 8 个子目录模块库（逐个依次进行），按住左键不放并拖拽到"Untitled"模型窗口下，建立名为 simulink30.mdl 的 M 文件，如图 3-4 所示。用鼠标左键双击其中任何一个图标，即可打开相应的模块库（图 3-5～图 3-12）。

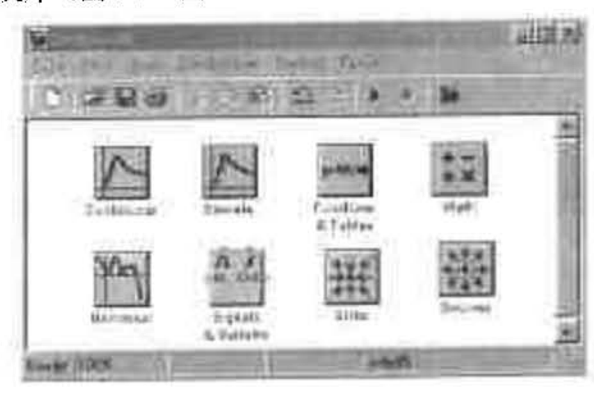


图 3-4　SIMULINK 3.0 的 8 类基本模块库

现在来分别介绍各类基本模块库中的标准功能模块图标及其功能。以下图 3-5～图 3-12 所示为 SIMULINK 3.0 模块库八个模块的介面、模块英文名与图标信息，表 3-1～表 3-8 所列信息是标准功能模块名称及其功能。

3.2.2.1 连续系统模块库（Continuous）及其用途

连续系统模块库及其图标如图 3-5 所示。图中显示有工具栏与状态栏，在介绍其他模块库及其图标的图中将隐藏工具栏与状态栏。MATLAB5.3 的连续系统模块库有 8 个标准基本模块。表 3-1 列出了这个子模块库中所有标准基本模块的名称与用途的简单介绍。

表 3-1　连续系统模块的名称与用途

模 块 名 称	模 块 用 途
Derivative	对输入信号进行微分
Integrator	对输入信号进行积分
Memory	输出本模块上一步的输入值
State-Space	建立一个线性状态空间模型
Transfer Fcn	建立一个线性传递函数模型
Transport Delay	对输入信号进行给定的延迟
Varible Transport Delay	对输入信号进行不定量的延迟
Zero Pole	以零极点形式建立一个传递函数模型

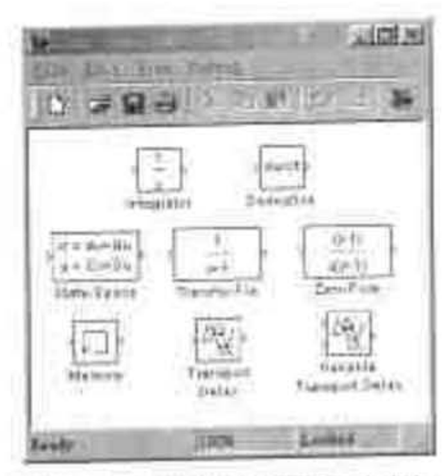


图 3-5 SIMULINK 的连续系统模块库

3.2.2.2 离散系统模块（Discrete）及其用途

离散系统模块库及其图标如图 3-6 所示。MATLAB5.3 的离散系统模块库有 8 个标准基本模块。表 3-2 列出了离散系统模块库中所有基本模块的名称与用途的简单介绍。

表 3-2 离散系统模块的名称与用途

模块名称	模块用途
Discrete Filter	建立离散（IIR 和 FIR）滤波器
Discrete State-Space	建立一个离散状态空间模型
Discrete Transfer Fcn	建立一个离散传递函数
Discrete Zero-Pole	建立一个零极点形式离散传递函数
Discrete-Time Integrator	对一个信号进行离散时间积分
First-Order Hold	建立一阶采样保持器
Unit Delay	对采样信号保持，延迟一个采样周期
Zero-Order Hold	建立零阶保持器

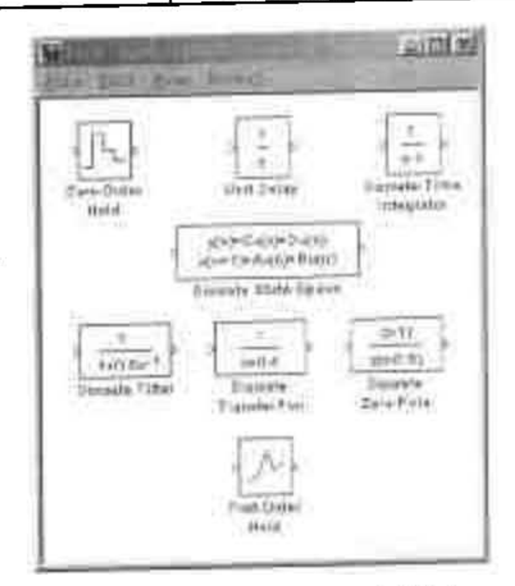


图 3-6 SIMULINK 的离散系统模块库

3.2.2.3　数学运算模块（Math）及其用途

数学运算模块库及其图标如图 3-7 所示。MATLAB5.3 的数学运算模块库有 20 个标准基本模块。表 3-3 列出了数学运算模块库中所有标准基本模块的名称与用途的简单介绍。

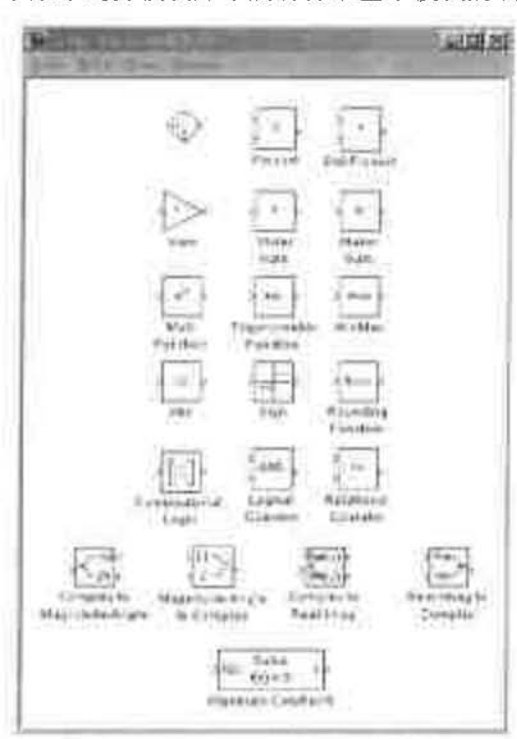


图 3-7　SIMULINK 的数学运算模块库

表 3-3　数学运算模块的名称与用途

模 块 名 称	模 块 用 途
Abs	求绝对值或求模（对复数）
Algebraic Constraint	强制输入信号为零
Combinnatorrial Logic	建立逻辑真值表
Complex to Magnitude-Angle	求复数的幅值与相角
Complex to Real-Imag	求复数的实部与虚部
Dot Product	求点积（内积）
Gain	对输入信号乘上一个常数增益
Logical Operator	逻辑操作符
Magnitude-Angle to Complex	由幅值与相角求复数
Math Function	数学运算函数
Matrix Gain	对输入信号乘上 一个矩阵增益
MinMax	求极大与极小值
Product	对输入信号求积或商
Real-Imag to Complex	由实部与虚部求复数
Relational Operator	比较操作符
Rounding Function	取整函数
Sign	取输入的正负符号
Slider Gain	以滑动形式改变增益
Sum	对输入信号求代数和
Trigonometric Function	三角函数

3.2.2.4 信号与系统模块（Signals & System）及其用途

信号与系统模块库及其图标如图 3-8 所示。MATLAB5.3 的信号与系统模块库有 26 个标准基本模块。表 3-4 列出了信号与系统模块库中所有基本模块的名称与用途的简单介绍。

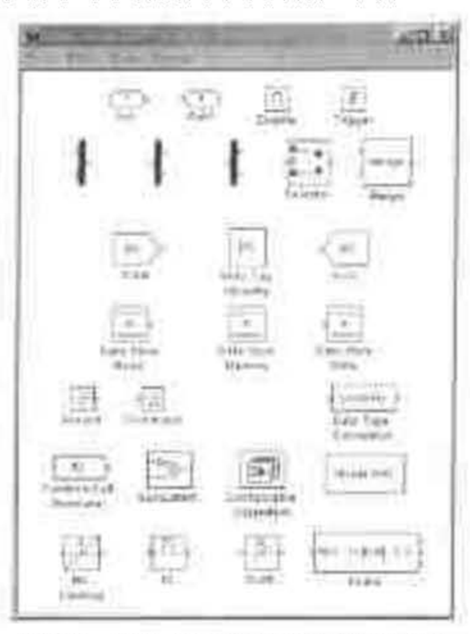


图 3-8 SIMULINK 的信号与系统模块库

表 3-4 信号与系统模块的名称与用途

模块名称	模块用途
Bus Selector	从输入中选择信号
Configurable Subsystem	从用户指定的模块库里选择任意模块
Data Store Memory	为数据存储器定义内存区域
Data Store Read	从指定的数据存储器中读数据
Data Store Write	写数据到指定的数据存储器
DataType Conversion	数据类型转换
Demux	将向量分为标量或小向量
Enable	在子系统中建立使能子系统
From	接受 Tag 标记相同的 Goto 模块的信号
Function-Call Generator	函数调用发生器
Goto	将信号输送到标记 Tag 相同的 From 模块
Goto Tag Visibility	定义 Goto 模块标记 Tag 的有效范围
Ground	使未连接的输入端接地，输出为 0
Hit Crossing	检测输入信号的零交叉点
Ic	设置信号初始值
In1,Out1	为模型或子系统提供输入、输出端口
Merge	将输入信号合并为标量输出信号
Modle Info	显示模型的修改信息
Mux	将向量或标量组合成大向量
Probe	检测连线的宽度、采样时间与复数信号标记
Selector	选择输入的元素
Subsystem	空的子系统
Terminator	接到未连接的输出端，终止输出信号
trigger	在子系统中建立触发子系统
Width	检测输入信号的宽度

3.2.2.5 非线性系统模块（Nonlinear）及其用途

非线性系统模块库及其图标如图 3-9 所示。MATLAB5.3 的非线性系统模块库有 10 个标准基本模块。表 3-5 列出了这个模块库中所有基本模块的名称与用途的简单介绍。

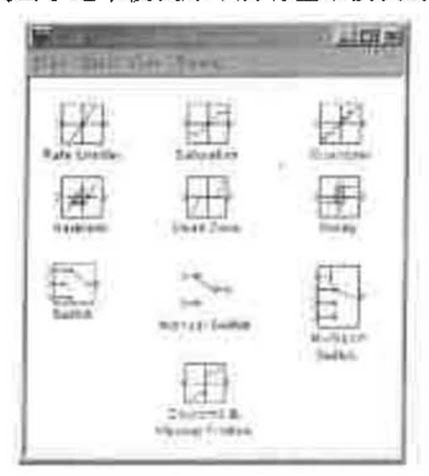


图 3-9 SIMULINK 的非线性系统模块库

表 3-5 非线性系统模块的名称与用途

模 块 名 称	模 块 用 途
Backlash	在输出不变区中输出不随输入变化而改变，在输出不变区外输出随输入成正比变化
Coulombic & Viscous Friction	在原点不连续，在原点以外具有线性增益
Dead Zone	提供一个死区特性
Manual Switch	手动开关
Multiport Switch	在多输入中选择一个输出
Quantizer	对输入进行阶梯状量化处理
Rate Limiter	限制信号的变化速率不超过规定的限制值
Relay	在正负限定值输出的带有滞环的继电特性
Saturation	对输出信号进行限幅的饱和特性
Switch	当第二个输入端信号大于临界值时，输出第一个输入端的信号，否则输出第三个输入端的信号

3.2.2.6 输出模块（Sinks）及其用途

输出模块库及其图标如图 3-10 所示。MATLAB5.3 的输出模块库有 6 个标准基本模块。表 3-6 列出了这个模块库中所有标准基本模块的名称与用途的简单介绍。

表 3-6 输出模块的名称与用途

模 块 名 称	模 块 用 途
Display	实时数字显示
Scope	在仿真过程中显示信号在类似示波器的窗口内
Stop Simulation	当输入不为零时停止仿真
To File	把数据输出到文件中
To Workspace	把数据输出到工作面上定义的一个矩阵中
XY Graph Scope	在 MATLAB 图形窗口显示信号的 X-Y 二维图形

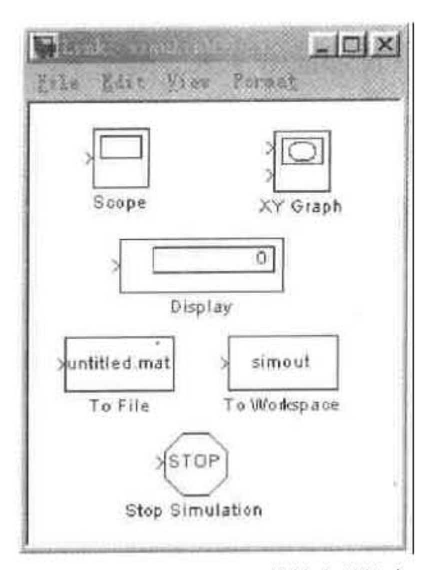


图 3-10　SIMULINK 的输出模块库

3.2.2.7　输入源模块（Sources）及其用途

输入源模块库及其图标如图 3-11 所示。MATLAB5.3 的输入源模块库有 16 个标准基本模块。表 3-7 列出了输入源模块库中所有基本模块的名称与用途的简单介绍。

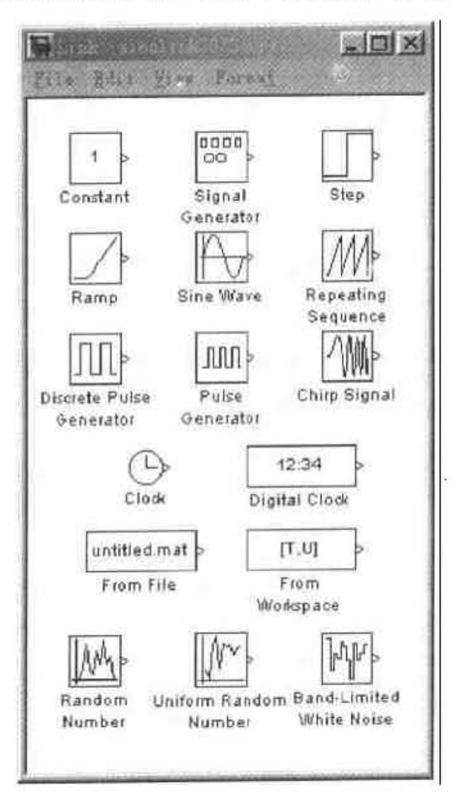


图 3-11　SIMULINK 的输入源模块库

表 3-7　输入源模块的名称与用途

模 块 名 称	模 块 用 途
Band-Limited White Noice	带限白噪声
Chip Signal	产生一个频率不断变化的正弦波信号
Clock	显示当前仿真时间
Constant	生成一个常值
Digital Clock	在规定的采样间隔显示当前仿真时间
Discrete Pulse Generator	离散脉冲信号发生器（与采样时间有关）
From File	从文件中读数据
From Workspace	从当前工作空间定义的矩阵中读数据
Pulse Generator	固定时间间隔的脉冲发生器
Ramp	线性增加或减小的信号（斜坡信号）
Random Number	正态分布的随机信号
Repeating Sequence	产生规律重复的线性信号（如锯齿形）
Signal Generator	产生各种不同的波形
Sine Wave	产生一个正弦波
Step	产生一个阶跃函数
Uniform-Limited White Noise	平均分布的随机信号

3.2.2.8　函数与表模块（Function & Tables）及其用途

函数与表模块库及其图标如图 3-12 所示。MATLAB5.3 的 连续系统模块库有 5 个标准基本模块。表 3-8 列出了这个模块库中所有标准基本模块的名称与用途的简单介绍。

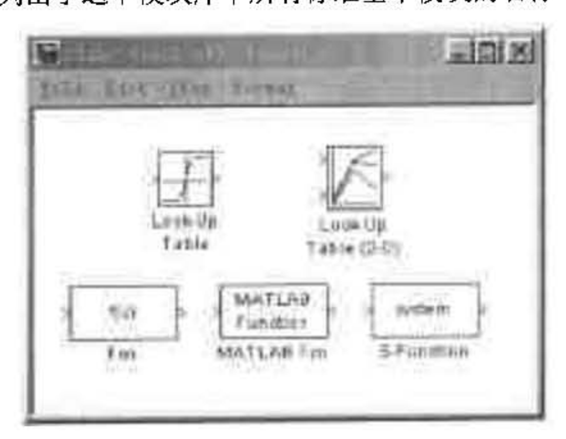


图 3-12　SIMULINK 的函数与表模块库

表 3-8　函数与表模块的名称与用途

模 块 名 称	模 块 用 途
Fcn	自定义函数运算
Look-up Table	建立输入信号的查询表
Look-Up Table（2D）	建立两维输入信号的查询表
MATLAB Fcn	用 MATLAB 现有函数运算
S-Function	调用 S 函数的程序

3.2.3 SIMULINK 模型窗口

当按下【新建】模型文件按钮时，就弹出无标题名称的“Untitledl”新建模型窗口（图 3-13）。当建立的模型文件命名后，标题“Untitledl”就改变为文件的名称，这个窗口即为模型窗口。MATLAB 规定模型文件即动态结构图模型的文件扩展名（或称为后缀）为“.mdl”。文件命名时不需要写入扩展名，MATLAB 会自动添加上去。

模型窗口第一行是模型窗口的主菜单，第二行是模型窗口的工具栏，最下方是状态栏。在工具栏与状态栏之间的大窗口是模型建立（画图）、修改、仿真的操作平台，这是一个非常重要的地方。SIMULINK 模型窗口主菜单与工具栏是 SIMULINK 仿真操作的重要技术内容。

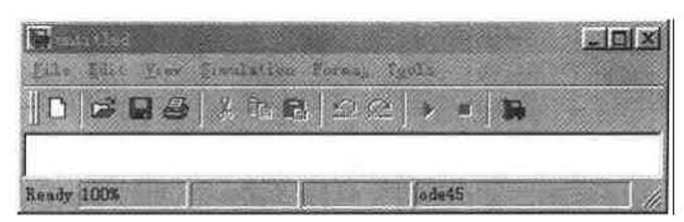


图 3-13 SIMULINK 的模型窗口

3.2.3.1 模型窗口的菜单

SIMULINK 3.0 模型窗口的条形主菜单有 File （文件）、Edit（编辑）、View （查看）、Simulation（仿真）、Format （格式设定） 与 Tools（仿真）等 6 项菜单选项。条形主菜单还可根据 SIMULINK 窗口的宽或窄自动排列成一行或两行。

6 项主菜单项都有其下拉菜单，每一个菜单项都是一个命令，只要用鼠标选中，即可执行菜单项命令规定的操作。以下是各个菜单项命令的等效快捷键（如果有的话）及功能。

1.【File】文件菜单

New	Ctrl+N	创建新的 SIMULINK 窗口
Open	Ctrl+O	打开已经存在的 SIMULINK 窗口
Close	Ctrl+W	关闭当前的 SIMULINK 窗口
Save	Ctrl+S	保存当前的模型文件（路径、子目录、文件名都不变）
Save As		将模型文件另外保存（改变路径或子目录或文件名）
Model Propeties		模型属性
Print	Ctrl+P	打印模型文件
Print Setup		打印设置
Exit MATLAB	Ctrl+Q	退出 MATLAB

2.【Edit】编辑菜单

Undo	Ctrl+Z	撤销前次操作
Redo	Ctrl+Y	恢复前次操作
Cut	Ctrl+X	剪切当前选定的内容，并放在贴板上
Copy	Ctrl+C	将当前选定的内容拷贝到粘贴版上

Paste	Ctrl+V	将粘贴版上的内容粘贴到当前光标所在位置
Clear	Delete	清除选定的内容
Select All	Ctrl+A	选择整个窗口
Copy Model		将模型拷贝到剪贴版上
Signal Propeties		信号属性
Block Propeties		模块属性
Create Subsystem	Ctrl+G	创建子系统模块
Mask Subsystem	Ctrl+M	封装子系统模块粘
Look Under Mask	Ctrl+U	详细察看封装模块的内部结构
Go To Library Link	Ctrl+L	恢复函数库模块与参考模块之间的联系
Break Library Link		切断函数库模块与参考模块之间的联系
Unlock Library		解锁函数模块库（使之能耗修改其中的模型）
Update Diagram	Ctrl+D	更新模型框图的外观

3.【View 】查看菜单

Toolbar	显示或隐藏工具栏
Statebar	显示或隐藏状态栏
Model Browser	显示模型浏览器
Block Data Tips	当鼠标位于模块上方时显示模块内部数据
Show Library Browser	立即显示模块库浏览器
Zoom In	放大模型显示比例
Zoom Out	缩小模型显示比例
Fit System To View	自动选择最合适的显示比例
Normal	以正常工作比例（100 %）显示模型

4.【Simulation 】仿真菜单

Start（ Pause）	Ctrl+T	启动或暂停仿真
Stop		停止仿真
Parameters	Ctrl+E	设置仿真参数
Normal		正规标准仿真
External		外部模型仿真

5.【Format】 仿真格式设定

Font		设置字体
Flip Name		模块标题名称上下换位
Show（Hide）Name		显示或隐藏模块标题名称
Flip Block	Ctrl+F	将功能模块图反向旋转 180°
Rotate Block	Ctrl+R	将功能模块顺时针旋转 90°
Show（Hide）Drop Shadow		显示或隐藏模块的阴影
Show（Hide）Port Labels		显示或隐藏子系统输入输出字符与空框或有示意图框
Foreground Color		设置前景颜色
Background Color		设置背景颜色

Screen Color	设置屏幕颜色
Sample Time Colors	给不同采样时间序列填加颜色
Wide Vector Lines	用宽信号线表示向量信号
Cector Line Widths	设置向量信号的宽度
Port Data Types	标明端口数据的类型

6.【Tools】工具菜单

Linear Analysis	线性化分析工具
Report Gelerator	模型文件的设置文件清单
RTW Build	RTW 构造生成器
RTW Options	RTW 的标签页
External Model Control Panel	外部模式控制板

3.2.3.2 模型窗口工具栏

模型窗口中主菜单下面是工具栏（图 3-14），工具栏有 12 个按钮，与 MATLAB 命令窗口的工具栏相类似，也是 SIMULINK 为使用者提供的模型窗口部分常用菜单项命令的快捷按钮。熟练使用工具按钮肯定能使操作更方便、更快捷高效。



图 3-14 SIMULINK 模型窗口的工具栏

图 3-14 所示的 SIMULINK 模型窗口工具栏自左到右的 12 个按钮，其功能分述如下：

第 1 个■：单击该按钮将打开一个新模型文件，相当于执行在主菜单项【File】选中的【New】命令；

第 2 个■：单击该按钮将打开一个已存在的模型文件，相当于执行在主菜单项【File】选中的【Open】命令；

第 3 个■：单击该按钮将保存模型文件，相当于执行在主菜单项【File】选中的【Save】命令；

第 4 个■：单击该按钮将打印模型文件，相当于执行在主菜单项【File】选中的【Print】命令；

第 5 个■：单击该按钮将选中的模型文件剪切到粘贴板上，相当于执行在主菜单项【Edit】选中的【Cut】命令；

第 6 个■：单击该按钮将粘贴板上的模型文件复制到指定位置，相当于执行在主菜单项【Edit】选中的【Copy】命令；

第 7 个■：单击该按钮将粘贴板上的文本内容粘贴到模型窗口中的指定位置，相当于执行在菜单项【Edit】选中的【Paste】命令；

第 8 个■：单击该按钮将撤销前次操作，相当于执行在主菜单项【Edit】选中的【Undo Delete】命令；

第 9 个■：单击该按钮将重复前次操作，相当于执行在主菜单项【Edit】选中的【Redo Delete】命令；

第 10 个▒：单击该按钮将启动或暂停仿真，相当于执行在主菜单项【Simulation】选中的【Start/ Pause】命令；

第 11 个▒：单击该按钮将将停止仿真，相当于执行在主菜单项【Simulation】选中的【Stop】命令；

第 12 个▒：单击该按钮将打开 SIMULINK 库浏览器，相当于执行在主菜单项【View】选中的【Show Library Browser】命令。

3.3 用 SIMULINK 建立系统模型

上一节介绍了 SIMULINK 的 8 类基本模块（图 3-3），这一节要介绍如何用 SIMULINK 建立系统建模型，即如何操作模块库提供的功能模块以及如何建立控制系统结构图模型。

SIMULINK 完全采用标准模块方框图的拷贝方法来构造动态系统的结构图模型。系统结构图模型的创建过程就是从 SIMULINK 模块库中选择所需要的标准功能模块，不断拷贝到模型窗口“untitled”里，再用 SIMULINK 的特殊连线方法把多个标准模块连接成描述（或代表）控制系统实际结构的方框图模型的过程。在 SIMULINK 环境中绘制模型结构图，并不很复杂，只是依赖于鼠标操作。鼠标的不同操作体现在光标的形状上，在缺省状态下鼠标是一个箭头，但在图形的操作过程中鼠标呈现不同的形状。

3.3.1 模型窗口

“untitled”窗口即无标题的空白窗口（图 3-13），是 SIMULINK 仿真工具用来绘制控制系统结构图模型的空白设计区，对控制系统创建的结构图模型（也叫模型方框图或方框图模型）就是在这个空白区里完成的。所以“untitled”窗口也叫做“方框图窗口”或者称“模型窗口”。建立 SIMULINK 模型就是在模型窗口里建立一个扩展名为“.dml”的文件，当然就得要先打开“untitled”窗口，尔后再在其中建立.dml 文件。模型创建完成后，需要给已建模型取一个见名知义的文件名并加以保存，那么“untitledl”无标题的空白窗口就变成了有名称的模型窗口了。

打开“untitled”模型窗口通常有四种方法：

（1）直接从 MATLAB 命令窗中选取【File】中【New】子菜单的【Model】命令，MATLAB 会打开一个新的“untitled”模型窗口。

（2）在 MATLAB 命令窗口下输入 simulink 命令，打开 SIMULINK 模块库浏览器窗口，然后再按【▒】按钮创建一个新“untitled”模型窗口。

（3）在 MATLAB 指令窗口下左键单击“New Simulink Model”（▒）按钮，再按【▒】按钮创建一个新的“untitled”模型窗口。

（4）如果方框图模型已经存在，那么在 MATLAB 指令窗下直接键入模型文件名字（不带扩展名），便会直接打开该模型方框图的模型窗口。用户可以对它直接进行编辑、修改和仿真。还可以在已打开的模型窗口里按照第（2）项再创建一个新模型窗口或按照第（4）项再打开已有的方框图模型。

3.3.2 模块的查找与选择

SIMULINK 3.0 提供的 8 类基本模块库存放着各种标准模块，所以模块的查找必须首先进入 SIMULINK 模块库浏览器窗口（图 3-3）。进入的方法，可以参见打开“untitled”模型窗口的 4 种方式中的前 3 种。

其次，对上节介绍的 SIMULINK 8 类模块库的组成及其中标准模块的名称、图标与功能应有大致的了解与熟悉，以便能够及时查找到所需要的模块。

需要指出的是，SIMULINK 的老版本，系统本身就有类似图 3-4～图 3-12 的模块库窗口界面。而 SIMULINK 3.0 只有模块库浏览器窗口（图 3-3），操作图 3-3 还不能直接打开模块库。图 3-4 的模块库窗口界面是作者制作的，不是系统给出的。建议使用 MATLAB 5.3 的读者为了使用的方便，不妨试一试按 3.2 节的方法制作如图 3-4 所示的 SIMULINK 3.0 模块库窗口。有了如图 3-4 所示的模块库窗口，寻找标准模块就会变得很方便了。如果没有图 3-4，就得按以下的方法打开模块库。

用 SIMULINK 3.0 的模块来构造系统的结构图模型，实际上是指用模块的图标来连接成模型结构图。SIMULINK 3.0 模块的使用与老版本模块的使用是不同的。在图 3-3 所示的模块库浏览器窗口里是看不到模块图标的，所有的模块都是带阴影的菱形标识（图 3-15），这些菱形标识位于 8 个基本模块目录之下的第三层。如果要选用某一个模块，必须先要以英语名字查找到其位置，再将模块菱形标识从库浏览器窗口里用鼠标选中并拖拽到“untitled”模型窗口中，菱形标识才会变成模块图标。



图 3-15 模块库中菱形标识的标准模块

其他所有模块都可以按以上方法复制到用户的模型窗口中。

至于模快正确的选择，则取决于对系统实际构成的各个环节的性质、特点与作用的深入了解，以及对于 SIMULINK 的 8 类模块库中各个标准模块的功能、使用方法的熟悉与掌握。

模块的使用请参看以下例 3-15 与例 3-16。

3.3.3 模块的拷贝、移动与删除

1. 模块选定（选中）

模块选定（即选中）操作是许多其他操作（如删除、剪切、拷贝）的“前导性”操作。选中模块的方法有两种：

（1）用鼠标左键单击待选模块，模块四个角处出现小黑块，表示已经选中。

（2）如果选择一组模块，可以按住鼠标左键拉出一个矩形虚线框，将所有待选模块框在其中，然后松开左键，则矩形里所有的模块，每个模块四个角处都出现小黑块，表示所有模块同时被选中。

关于模块的选取还须说明两点：

（1）如果在选中的模块图标上再次单击左键，模块四个角处出现的小黑块则会消失，表示取消了对该模块的选取。

（2）如果想选取多个模块对象，但是用拖拽方框的方式又会选取到我们不想要的模块对象，此时可以按住【shift】键，再用按住鼠标左键来拖拽一个矩形虚线框，要一个就框一个，不要的就不框，这样一个一个地选取。

2. 模块的拷贝（复制）

模块的拷贝包括：一是从模块库中将标准模块拷贝到模型窗口“untitled”里；二是在“untitled”模型窗口里将模块再拷贝。

从模块库中拷贝标准模块的操作方法是：在模块库中将鼠标箭头尖指向待选模块，并用鼠标左键单击之，待选模块四个角处出现小黑块，表示已经选中，按住鼠标左键不放将所选模块拖拽到“untitled”模型窗口里的目标位置，松手放开鼠标左键，在“untitled”模型窗口里某个位置上就有一个与待选模块完全相同的模块图标，即已完成模块从库中的拷贝。对于同一标准模块或者不同的模块均可以多次重复这样的拷贝操作。

在“untitled”模型窗口里拷贝模块的方法有三种。

（1）一种方法是首先选中待拷贝模块，运行【Edit】菜单中【Copy】命令；然后将光标移到将粘贴的地方，按一下鼠标左键；看到选定的模块恢复原状，在选定的位置上再运行【Edit】菜单中的【Paste】命令即可。新复制的模块和原装模块的名称也会自动编号，以资区别。

（2）另一种简单的复制操作是：先按下【Ctrl】键不放，然后将鼠标移到模块对象上，注意看鼠标指针，如果多了一个小小的“加号”，表示可以复制了。把鼠标拖拽到目的位置后，松开鼠标左键，便完成拷贝工作。

（3）还有一种更为特别的简单复制操作：用鼠标指向待拷贝模块对象，按住鼠标右键不放，将鼠标拖拽到目的地，放开鼠标右键，即可复制一个功能模块。

3. 模块的移动

模块移动操作非常简单：将光标置于待移动模块图标上，然后按住鼠标左键不放，将模块图标拖拽到目的地，放开鼠标左键，模块移动完成。注意，模块移动时它与其他模块的连线也随之移动。

4. 模块的删除和粘贴

对选中的模块的删除和粘贴可以操作：

（1）按【Delete】键，把选定模块删除。

（2）选择【Edit】菜单中的【Cut】命令后，便将选定模块移到 Windows 的剪粘贴板上，再用【Paste】命令重新粘贴。

5．改变模块对象的大小

用鼠标选择对象模块图标，再将鼠标移到模块对象四周的控制小块处，鼠标指针将会变成“↙”或“↘”或“↗”或“↖”形状，此时按住鼠标左键不放，拖拽鼠标，待对象图标大小符合要求时即放开鼠标左键，这样就可改变模块对象图标的大小。

6．改变模块对象的方向

一个标准功能模块就是一个控制环节。在绘制控制系统模型方框图即连接模块时，要特别注意模块的输入、输出口与各模块间的信号流向。在 SIMULINK 中，模块总是由输入端口接受信号，输入端口位于模块左侧；模块总是从输出端口发送（出）信号，输出端口位于模块右侧。但是在绘制反馈通道时则会有相反的要求，即输入端口不在模块左侧，那么输出端口就不在模块右侧。这种要求可以按以下操作步骤来实现：用鼠标选中模块对象，利用“untitled”的主菜单项【Format】下拉菜单的【Flip Block】或者【Rotate Block】命令，如果选择【Flip Block】或者直接按【Ctrl＋F】键，即可将功能模块旋转 180°；如果选择【Rotate Block】或者直接按【Ctrl＋R】键，即可将功能模块顺时针旋转 90°。

3.3.4 模块的连接

将构成一个控制系统的所有环节模块拷贝到“untitled”模型窗口里，这些模块图标在没有用信号线将其连接之前，并不描述一个控制系统，即不代表任何系统模型。必须用信号线将各个模块图标连接成能够描述一个控制系统的所谓系统模型。所以，要说明模块的连接还得首先介绍信号线的使用。

1．信号线的使用

信号线具有连接功能模块的作用。用鼠标箭头在“untitled”模型窗口里拖动，可以在模块的输入与输出之间直接连信号线。前面已经介绍过，为了连接两个模块，按住鼠标的左键，点击输入或输出端口，看到光标变为十字形以后，拖拽十字图形符号到另外一个端口，鼠标指针将变为双十字形状，然后放开鼠标左键。于是一根最简单的信号线即将两模块连接起来，带连线的箭头表示信号的流向。

对信号线操作时，如同对模块操作一样也须先选中（鼠标左键单击该线），被选中的信号线，其两端出现两个小黑块，那么，就可以对该信号线进行操作，诸如改变其粗细、对其设置标签，也可以把信号线折弯、分支，甚至于将该信号线删除等等。

2．向量信号线与线型设定

对于向量信号线，在“untitled”模型窗口里，选中的主菜单项【Format】下【Wide Vector Lines】的命令，对模型执行完【Simulation】下的【Start】命令或【Edit】下的【Update Diagram】命令之后，传输向量的线即会变粗。此即说明，变粗了的线段，表示该连接线上的信号为向量形式。

3．信号线设置标签

只要在信号线上双击鼠标左键，即可在该信号线的下部拉出一个矩形框，在矩形框内的光标处即可输入该信号线的说明标签，既可输入西文字符也可以输入汉字字符。标

签的信息内容如果很多，还可以按回车键换行输入。如果标签信息有错或者不妥也可以重新选中再编辑修改。

4．信号线折弯

对选中的信号线，按住【Shift】键，再用鼠标左键在要折弯的地方单击一下，在此处就会出现一个小圆圈，表示折点，利用折点就可以改变信号直线的形状。

对选中的信号线，将鼠标指到线段端头的小黑块上，直到箭头指针变为“O”，按住鼠标左键，拖拽线段，即可将线段以转直角的方式折弯。

如果不想以直角的方式折弯，则可以在线段的任一位置，按住【shift】键与鼠标左键，将线段以任意角度折弯。

5．信号线分支

对选中的信号线，按住【Ctrl】键，并在要建立分支的地方按住鼠标左键拉出即可。另外一种方法是：将鼠标指到要引出分支的信号线段上，如果按住鼠标右键拖拽鼠标，还可拉出分支线段。如果按住鼠标右键，拖拽鼠标可以拉出非直角线段。

6．信号线的平行移动

将鼠标指到要平行移动的信号线段上，按住鼠标左键不放，鼠标指针变为十字箭头形状，水平或者垂直方向拖拽鼠标移到目的位置，放开鼠标左键，信号线的平行移动即完成。

7．显示向量信号线上的信号数目

在“untitled”模型窗口主菜单的【Format】选项中运行【Vextor Line Widths】命令，模型框图的所有向量连接线上两段出现一个小阿拉伯数字，这个数字表示该线段内有多少个信号，从而就可以知道某一个模块有多少个输入、输出信号。重复使用上述命令，连接线上两段的小阿拉伯数字又会消失。

8．信号线与模块分离

将鼠标指针放在想要分离的模块上，按住【Shift】键不放，再用鼠标把模块拖拽到别处，即可以把模块与连接线分离。

9．信号线的删除

对选中（定）的信号线的删除操作非常简单，按【Delete】键，即可把选中的信号线删除。

【例 3-1】试绘制一阶系统进行阶跃响应实验的结构图模型。

【解】

绘制一阶系统结构图模型图的步骤如下：

（1）利用 SIMULINK 窗中的“”按钮创建一个新的“untitled”模型窗口。

（2）从输入源模块库（Sourse）、数学运算库（Math）、连续系统库（Continuous）、输出模块库（Sinks）中，分别用鼠标把阶跃信号发生器（Signal Generator）、传递函数（Transfer Fcn）、示波器（Scope）、求和器（Sum）四个标准功能模块先选中并拖拽到“Untitled”模型窗口中。各模块的位置如图 3-16 所示。

（3）图 3-16 中模块间的连线有两类：一类是“直”线，另一类是“折”线。先将模块间连直线，再连折线：把鼠标光标移到传递函数和示波器间连线的中点附近，按下鼠标右键，光标由箭头变为十字，往下拉动鼠标到适当位置后放开右键，屏幕上就出现

一条由连线中点引出的箭头线，再从此处开始拖拽鼠标向左画线到适当位置，再松开鼠标键。照此操作，直到整个“折线”绘制完成。

（4） 在反馈通路线的上部或下部双击鼠标左键，即可在该信号线旁拉出一个矩形框，在矩形框内的光标处输入“反馈线”，则该信号线的说明标签即告完成。

（5） 用鼠标选中求和器并双击左键，打开求和器的对话框，在“List of signs”栏里填入“|+-”字符，图 3-16 所示的图形：即绘制成功。

最后，对建立好的系统方框图模型还需命名后存盘。

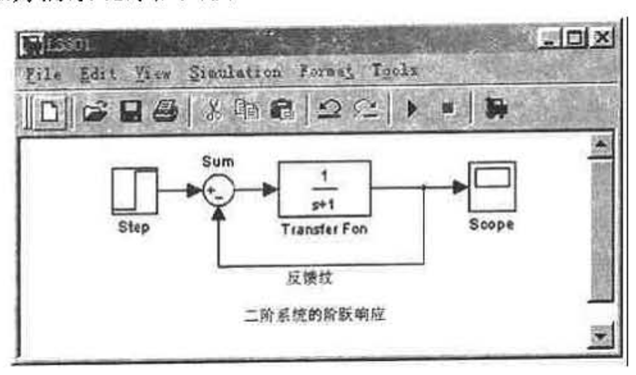


图 3-16　一阶系统结构图模型

3.3.5 模块标题名称的修改

模块的一般属性可以分为两方面：一是指模块的标题名称，另一个是指模块的内部参数设置。为解决实际工程问题的需要，那些被复制标准模块的标题名称和内部参数设置常常需作必要的修改，所以了解标题名称的修改方法是需要的。

模块标题名称是指标识模块图标的字符串，通常模块标题名称设置在模块图标的下方。也可以将模块标题名称设置在模块图标的上方（即在模型窗口中使用菜单【Format】下的命令【Flip Name】），甚至于还可以将模块标题名称隐藏起来（即在模型窗口中使用菜单【Format】下的【Hide（Show） Name】命令将模块标题名称隐藏或显示）。对用户所建模型窗中模块标题名称进行修改的具体方法如下：

（1）用鼠标左键单击功能模块的标题，在原模块标题外拉出一矩形框，按住鼠标左键对要修改的标题字符部分使之增亮反相显示。

（2）按回车键，反相显示的要修改的部分字符立即被删除，重新输入新的标题信息（中西文字符均可）。

（3） 然后用鼠标左键单击窗口中的任一地方，修改工作结束。

需要指出的是：如果重新输入新的标题信息内容很多（很长），可按回车键换行输入。

3.3.6 模块内部参数的修改

模块的第二个属性是指模块的内部参数设置。在模型窗口中，双击待修改参数模块的图标，便打开标准功能模块内部参数设置对话框，然后通过改变对话框相关栏目中的数据便可。下面通过具体示例来说明具体操作。

【例 3-2】对方框模型图 3-17 所示系统标注模块标题、进行参数设置并进行给定阶跃响应仿真。

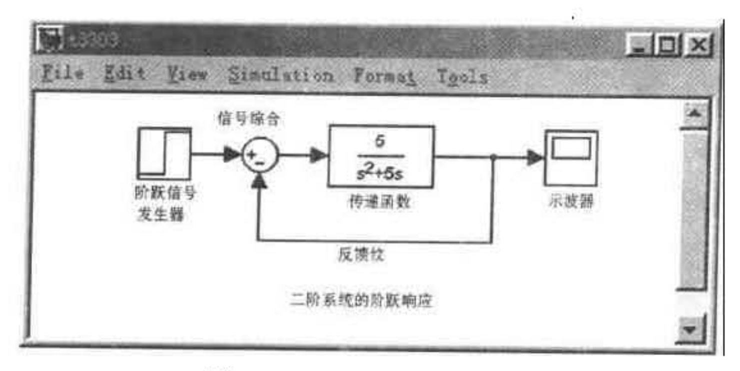


图 3-17　二阶线性系统模型

【解】

（1）对模块标题名称的修改

用鼠标左键选中并单击传递函数模块标题“Transfer Fcn”，将原标题字符删除重新输入汉字字符“传递函数”。再同样选完示波器模块标题“Scope”，删除后再输入汉字字符“示波器”，如此依次将“Step”改为“阶跃信号发生器”，将“Sun”改为“信号综合”。最后再在窗口空白处点击鼠标左键，标题修改结束，系统如图 3-18 所示。

（2）对阶跃信号发生器参数的修改

用鼠标左键选中并双击阶跃信号发生器，就会弹出如图 3-18 所示的对话框。对话框中参数设置有四个：“Step time”为阶跃信号产生的时间；“Initial value”为阶跃信号初始值；“Final value”为阶跃信号终了值；“Sample time”为采样时间。可以填入数据，如图 3-18 所示，再依次点击【Apply】与【OK】按钮，即完成阶跃信号发生器参数的设定。

（3）对求和块输入极性的修改

用鼠标双击求和块（Sum），就会弹出如图 3-19 所示的对话框。

把“List of signs”栏中的缺省极性改成如图 3-19 所示形式，在“Icon shape”栏选择“round”，再依次点击【Apply】与【OK】按钮，原求和模块图标便自动改成图 3-19 所示形式。

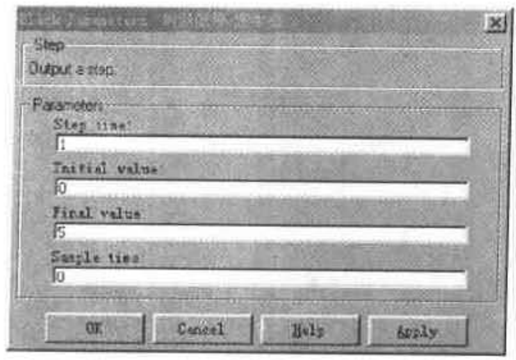


图 3-18　阶跃信号“Step”模块对话框

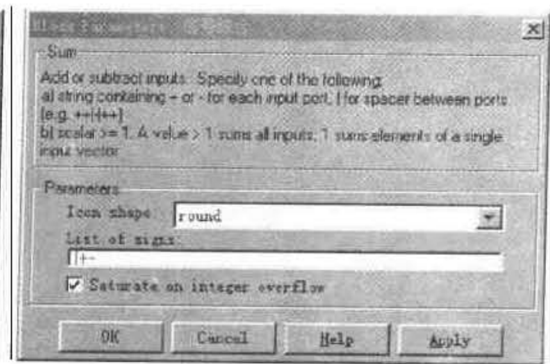


图 3-19　信号综合“Sum”模块对话框

（4）对传递函数模块参数的修改。

用鼠标双击传递函数模块图标，弹出如图 3-20 所示对话框。

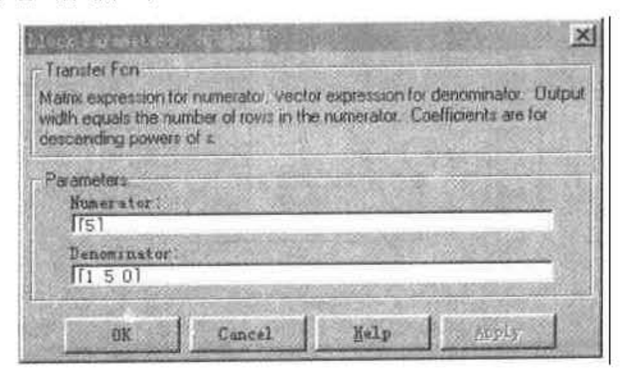


图 3-20 传递函数“Transfer Fcn”模块对话框

“Numerator”栏是传递函数的分子多项式系数向量。“Denominator”栏是传递函数的分母多项式系数向量。把这两栏中的原缺省值改填成图 3-20 所示的行向量，再依次点击按钮【Apply】与【OK】键，原传递函数模块图标中的函数表达式就自动变成图 3-17 中的样式。

值得指出的几点是：

①传递函数的分子、分母多项式系数行向量的输入，是按降幂排列的顺序从高（左）到低（右）依次输入，各项系数之间必须输入空格字符。

②如多项式缺项，必须将对应系数 0 输入，不能遗漏。

③在参数设置时，任何 MATLAB 工作内存中已有的变量、合法表达式、MATLAB 语句等都可以填写在设置栏中。

④模块图标的大小是可以用鼠标操作调整的。因此，假如传递函数表达式太长，原方框容纳不下，可以用鼠标把它拉到适当大小，使整个方框图标美观易读。

（5）模型系统的单位阶跃给定响应仿真

打开 t3303 模型窗口（图 3-17），先选中菜单【Simulation】的【Start】命令，即对模型系统进行单位阶跃给定响应仿真，再双击示波器图标即可得仿真结果曲线，如图 3-21 所示。

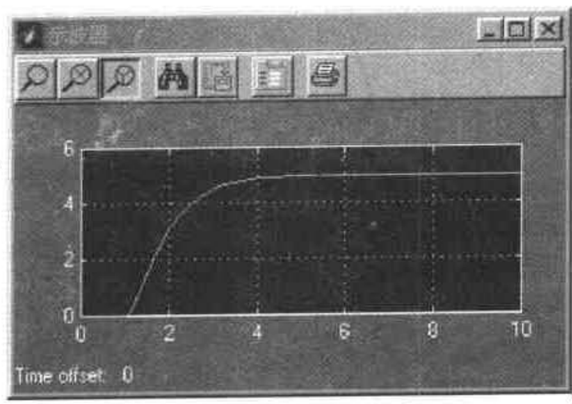


图 3-21 例 3-2 给定阶跃响应仿真曲线

3.3.7 创建模型的复原操作

Simulink 允许在创建模型的过程中，如果出现增减模块、增减线段、增减模型注解、编辑模块名称等等误操作，可以取消这个错误的操作。只要选择模型窗口里【Edit】菜单内的【Undo】命令即可。

也可以选择【Redo】命令来取消【Undo】命令，即恢复原操作。

3.3.8 系统模型标题名称的标注

在“untitled”模型窗口里，将鼠标指针指在窗口的空白处，双击鼠标左键，在模型窗口里鼠标指定的位置出现一个小方框，并且小方框内有文字光标在闪动，即可以在方框内给系统模型标注名称、标题或者作注解。

3.3.9 模型文件的保存与打开

编辑好一个模型后，可以在“untitled”模型窗口中选择【File】菜单中的【Save】命令将模型以原文件名存盘。模型是以 ASCII 码形式存储的 .mdl 文件，也就是 MATLAB 规定动态结构图模型的文件扩展名为“.mdl”。动态结构图模型文件名可以省略扩展名，系统会自动添加上去。文件包含了该模型的所有信息，既有这个数学模型的内涵，又有其外部方框图的可见形式。

也可以在“untitled”模型窗口中选择【File】菜单中的【Save As】命令将模型文件在设定的路径与设定的子目录下（最好的路径是 MATLABR11\WORK），以一个新命名的文件名称存盘。

必须特别指出的是，以某个文件名的.mdl 文件如果已经存在，再次以该文件名保存内容不同的文件时，新的文件内容将覆盖原文件内容，原文件内容是不能恢复的。

由 SIMUINK 创建的 .mdl 文件也可以在 MATLAB 命令窗口里用 type 指令查看其内容，图 3-22 所示为查看 L3302.mdl 模型对应的 MATLAB 程序的内容。

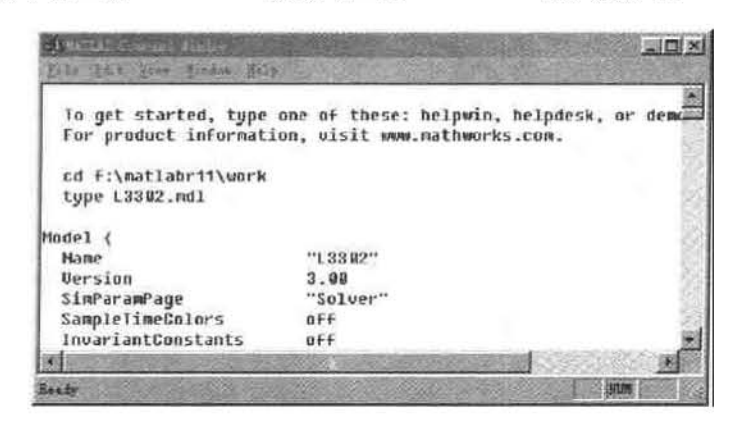


图 3-22 在 MATLAB 命令窗口查看模型内容

已经保存在计算机磁盘上的模型文件（.mdl 文件）可以用多种方法打开。可以点击库浏览器里或模型窗口里的“ ”图标；也可以使用模型窗口里【File】菜单的【Open】

子菜单命令；还可以在 MATLAB 命令窗口里直接输入欲打开模型文件的名字（要注意：一是不要带文件扩展名；二是必须注明模型文件所在的路径与子目录）等方法都能够打开模型文件。

3.3.10 模型框图的打印

SIMULINK 集成环境下建立的系统模型框图，可以用以下方法打印输出。在模型窗口里，利用【File】菜单下的【Print】命令或按下工具栏的“▓”按钮则打印当前活动窗口的框图，而不打印任何打开的 Scope（示波器） 模块。

3.3.11 SIMULINK 建模注意事项

为了更好地使用 SIMULINK，这里提出几点注意事项，请读者予以关注。

（1） 建模操作

建模时应把解决的问题思考清楚，首先在纸上画出草图，然后才在计算机上输入。其次，只有你用 SIMULINK 把所有的模块都拷贝到模型窗口里以后，才可以把这些模块连接起来。采用这些方法，将有助于减少打开文件所需要的时间，提高工作效率。

（2）模型的整理

模型结构要符合一般制图的规范，清晰、整齐有助于理解与阅读。除此之外，模型标题名称与模块名应简单明了；在模型中注释的信息，可解释模型中应该注意的问题或模型运行的条件、结果或各种发生与没有发生的事项等等，它有助于对模型的理解与阅读。

（3）关注仿真速度

①内存应足够大。一般来说，内存越大，仿真速度就越快。内存较小的机子，仿真时的数据需要分批调入内存。还有，在仿真没有结束时就得把一些中间结果写到磁盘上，以腾出内存空间，这样就大大减慢了仿真速度。

②合理选择仿真步长。将仿真步长或者采样周期设定得过小，虽可捕捉到仿真过程中的一些重要的细节，但是会产生许多不必要的输出，浪费许多时间，从而降低了仿真速度。所以应该选择或设定适当的仿真步长，既不得过大，也不要过小。

③避免出现代数环

有代数环的系统微分方程模型，其状态向量同类项没有合并，使得仿真运算速度变慢。因为代数环的解每步都需迭代进行，必然严重降低仿真速度，有时甚至使仿真无法进行。所以应尽可能避免出现代数环。

代数环是指当构成系统的环节含有构成代数环的模块类型时，所有模块都是由构成代数环的模块形成的一个闭合环路。这些构成代数环的每个模块，输入与输出都有代数关系，这些模块计有：增益模块、乘运算模块、求和模块、部分数学模块、部分非线性模块、D 非零的状态方程模块、分子分母同阶的传递函数模块、零极点数相等的零极点增益模块、积分模块的初始条件输入口等。

④程序中尽可能少用 MATLAB Fcn 模块。当模型中包含有 MATLAB Fcn 模块时，仿真每进行一步，就需要调用 MATLAB 解释程序，这样将大大降低仿真速度，因此最好尽可能不使用 MATLAB Fcn 模块，而使用 MATLAB 的基本运算函数。

3.4 SIMULINK 仿真算法的数学基础

3.4.1 微分方程初值问题的数值解法

SIMULINK 仿真算法的数学基础就是求常微分方程初值问题的数值解法。一个 n 阶连续系统可以用由 n 个积分器构成的模型来描述。

n 阶连续系统的计算机仿真，就是用数值解法来求解模型中的 n 个一阶微分方程的

$$\begin{cases} y' = \dfrac{\mathrm{d}\,y(t)}{\mathrm{d}\,t} = f\left(t, y\left(t\right)\right) & a \le t \le b \\ y\left(a\right) = y_0 \end{cases} \tag{3-1}$$

初值问题。

所谓数值解法，就是要逐个求出区间[a,b]内若干个离散点 $a \leqslant t_0 < t_1 < \cdots < t_n \leqslant b$ 处的近似值 $y(t_1)$， $y(t_2)$， …， $y(t_n)$。

理论上已经证明，只要函数 $f(t, y(t))$ 适当光滑，可以确认式（3-1）初值问题的解存在并且是唯一的。

微分方程的数值解法多种多样，这里只是介绍几种最基本解法的运算原理与相关概念，目的是为 SIMULINK 的仿真解算提供数学基础。

3.4.2 欧拉法

欧拉法是最简单也是最早的一种数值算法。将（3-1）中的微分方程两边进行积分，有：

$$\int_{t_i}^{t_{i+1}} \frac{\mathrm{d}\,y(t)}{\mathrm{d}\,t} = \int_{t_i}^{t_{i+1}} f\,(t, y\,(t))\,\mathrm{d}\,t \tag{3-2}$$

或

$$y\,(t_{i+1}) = y\,(t_i) + \int_{t_i}^{t_{i+1}} f(t, y\,(t))\,\mathrm{d}\,t \tag{3-3}$$

若不特别申明，一般设定离散点 t_0， t_1， …， t_n 是等距离的，即 $t_{i+1} - t_i = h$，其任意两点之间的距离 h 叫做计算步长或称为步距。

当 $t > t_0$ 时，把积分间隔取得足够小，使得在 t_i 与 t_{i+1} 之间的 $f(t, y(t))$ 可以近似地看成常数 $f(t_i, y(t_i))$。这样便得到用矩形面积来近似积分计算的公式：

$$y\,(t_{i+1}) \approx y\,(t_i) + f\,(t_i, y\,(t_i))\,(t_{i+1} - t_i) \tag{3-4}$$

或

$$y_{i+1} = y_i + h\,f_i,\quad i = 0,1,2,\cdots,n-1 \tag{3-5}$$

这就是欧拉公式。

欧拉公式的计算原理可以用图 3-23 来解释。

设 $i = 0$, 则有：

$$y(t_1) \approx y(t_0) + hf(t_0, y(t_0)) = y_0 + hf(t_0, y_0) = y_1 \tag{3-6}$$

式中，y_0 是 $t=t_0$ 时的初始值，$f(t_0, y_0) = \left.\frac{\mathrm{d}y}{\mathrm{d}t}\right|_{t=t_0}$ 是 $y^*(t)$ 在 (t_0, y_0) 处的斜率。从图 3-23 可以看出，在时间间隔（$t_1 - t_0$）内，曲线 $y(t)$ 是用 $y^*(t)$ 在 t_0 处的切线来近似的。在 t_1 处，显然 $y(t_1) \neq y^*(t_1)$。

反复使用式（3-5），便可得求一系列的近似解 $y(t_1)$，$y(t_2)$，…，$y(t_n)$。把这些点连接起来，便可得到如图 3-24 所示的一条折线，这条折线称为欧拉折线。

欧拉法又称矩形法或折线法。由于它比较简单，并且能够说明数值算法的基本原理与过程，所以它是数值积分解算的基础方法。但是由于它不够精确，现在已很少实际使用。

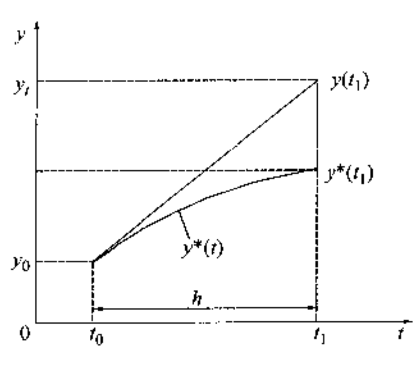


图 3-23　欧拉公式计算原理

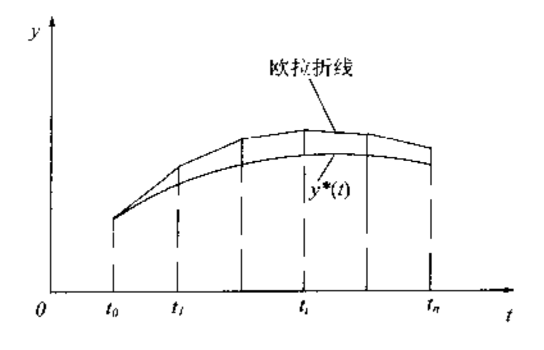


图 3-24　欧拉折线

3.4.3　数值积分运算的几个概念

为了全面了解数值积分运算，介绍几个基本概念。

1．截断误差

对欧拉公式表达式：

$$y(t_{i+1}) \approx y(t_i) + f(t,\ y(t_i))(t_{i+1} - t_i)$$

若将 $f(t, y(t))$ 展开成台劳级数（注意 $t_{i+1} - t_i = h$），则有：

$$y(t_{i+1}) = y(t_i + h) = y(t_i) + y'(t_i) + \frac{1}{2!}h^2 y''(t_i) + \cdots =$$

$$y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2!}f'(t_i, y(t_i)) + \cdots \tag{3-7}$$

比较两者可以看出，欧拉公式就是精确解式（3-5）截去第三项及其以后各项所得到的近似值，这样引起的误差称为截断误差。欧拉法的截断误差为

$$y(t_{i+1}) - y_{i+1} = O(h^2) \tag{3-8}$$

不同的数值方法，截误差也不同，因此常常用截误差来表示各种数值方法的精度。当某数值方法的截断误差为 $O(h^{p+1})$时，我们称它具有 p 阶精度。因此欧拉法为一阶精度的算法。

2．舍入误差

由于计算机的字长为有限的位数，故计算机运算的结果只能是有限位，在有限位以后的数值只能按某种规则舍掉，这样引起的误差叫做舍入误差。产生舍入误差的原因除计算机的字长外，还有以下因素：

（1）程序中数值运算的次序。

（2）运算采用定点或是浮点方式。

（3）计算机所用的数字系统。

（4）计算 $f(t,y(t))$ 所用的子程序的精度等。

3．单步法和多步法

求解初值问题数值方法的方式为步进方式，即从最初一点或几点出发，每一步都根据 y_i 这一点或前面几点 y_{i-1} ，y_{i-2} …的值来计算新的 y_{i+1} 的值，这样反复进行，逐步向前推进。

当从 t_i 推进到 t_{i+1} 只需用到 t_i 时刻数据的算法，称为单步法。如果从 t_i 推进到 t_{i+1} 需要用到 t_i 以及过去时刻 t_{i-1}，t_{i-2} … 数据的算法， 称为多步法。

4．显式和隐式

在计算 y_{i+1} 时所用到的数据均已解算出来并能查看得到，这种算法称为显式算法。若是在算式中隐含有未知量 y_{i+1} 时，则称为隐式算法。

显然，单步法和显式算法在计算上比多步法和隐式算法方便。但有时由于精度或者是计算稳定性的原因，必须采用隐式算法。

5．数值稳定性

采用数值方法求解稳定的微分方程时，应该保持原系统稳定的特征。但是，在计算机逐次计算时，初始数据的误差及计算过程的舍入误差对后面的计算结果将会产生影响。所谓稳定性问题就是指误差的积累是否能够受到控制的问题。如果计算结果对初值误差以及计算误差不敏感的话，则相应的计算方法是稳定的，否则是不稳定的。

3.4.4 阿达姆斯法（ 梯形法 ）和预估-校正法

在欧拉法的基础上稍作改进，就可得到梯形法，梯形法又叫做阿达姆斯（Adams）法 。

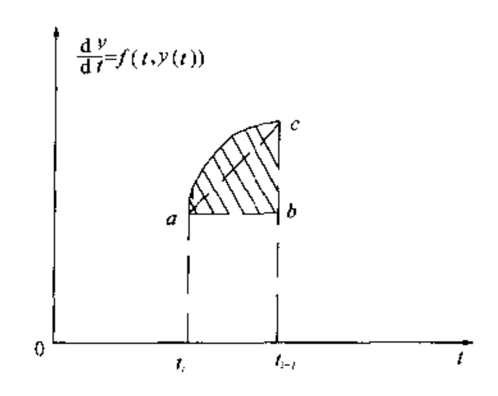


图 3-25　梯形法的几何意义

已经说明，欧拉公式的积分 $\int_{t_i}^{t_{i+1}} f(t, y(t))\mathrm{d}t$ 是用矩形面积 $hf(t_i, y(t_i))$ 来近似的。从图 3-25 可以看出，用矩形面积 $t_i abt_{i+1}$ 代替积分，其误差就是阴影部分。如果将积分值用 $t_i act_{i+1}$ 的梯形面积来近似，其误差显然减小了很多。为此，利用梯形公式便有：

$$\int_{t_i}^{t_{i+1}} f(t, y(t))\mathrm{d}t = \frac{h}{2}\left[f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1}))\right] \tag{3-9}$$

这个梯形公式叫做二阶隐式阿达姆斯法公式。它表明被积函数用 $y(t)$ 在 t_i 和 t_{i+1} 两点的斜率的平均值

$$\frac{h}{2}\left[f(t_i,\ y(t_i)) + f(t_{i+1},\ y(t_{i+1}))\right] = \frac{h}{2}(f_i + f_{i+1}) \tag{3-10}$$

来计算，其精度比用其中一点的斜率计算来得高。于是式 （3-3）为梯形法计算公式：

$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1}))\right] = y_i + \frac{h}{2}(f_i + f_{i+1}) \tag{3-11}$$

由于式（3-10）右边包含未知量 y_{i+1}，所以每一步都必须通过迭代求解。每一步迭代的初值通常采用欧拉法来计算。因此，梯形法每一步的迭代公式为：

$$y_{i+1}^{(0)} = y_i + hf(t_i,\ y_i) \tag{3-12}$$

$$y_{i+1}^{(k+1)} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{(k)})\right] \tag{3-13}$$

由于梯形法计算每步都要进行多次迭代，计算量特别大。为了简化计算，作为梯形法的一个特例，只对式（3-13）进行一次迭代便认为得到了式（3-11）的近似值。按照这样的考虑进行处理，计算公式便成为：

$$y_{i+1}^{(0)} = y_i + hf(t_i, y_i) \tag{3-14}$$

$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i,\ y_i) + f(t_{i+1},\ y_{i+1}^{(0)})\right] \quad k = 0,1,2,\cdots \tag{3-15}$$

这种一次迭代的方法叫做预估-校正方法。公式（3-14）是显式欧拉公式，用它算出的 $y_{i+1}^{(0)}$ 作为 y_{i+1} 的预估值，然后用式（3-15）对 $y_{i+1}^{(0)}$ 进行校核以得到更准确的近似值 y_{i+1}。

依照欧拉法确定截断误差的方法，可以求得梯形法和预估-校正法的截断误差为 $O(h^3)$，它们具有二阶精度。

除二阶隐式阿达姆斯法公式即梯形法公式外，常用的阿达姆斯公式还有：

（1）四阶精度显式

$$y_{i-1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) \tag{3-16}$$

（2）四阶精度隐式

$$y_{i+1} = y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}) \tag{3-17}$$

（3）四阶精度的预估-校正公式

$$y_{i+1}^{(0)} = y_i + \frac{h}{12}\left(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}\right) \tag{3-18}$$

$$y_{i+1} = y_i + \frac{h}{24}\left(9f_{i+1}^{(0)} + 19f_i - 5f_{i-1} + f_{i-2}\right) \tag{3-19}$$

阿达姆斯方法是线性多步算法，适用于有光滑解的非刚性问题。其最大的优点是每步只需计算一次 f 函数值（用预估-校正法时为两次），计算量小。缺点是不能自动开始，开头几步必须用其他方法计算，而且改变步长困难。此外，阿达姆斯方法是条件稳定的，即使是用隐式算法也不宜解刚性问题。

3.4.5 龙格-库塔法

重写台劳公式：

$$y_{i+1} = y_i + hy_i' + \frac{h^2}{2!}y_i'' + \cdots + \frac{h^p}{p!}y_i^{(p)} + O\left(h^{p+1}\right) \tag{3-20}$$

由式（3-20）可以看出，提高截断误差的阶次，便可提高计算精度。若是直接采用台劳公式，由于计算各阶导数相当麻烦而不可取。为了解决精度问题，龙格首先提出了间接使用台劳级数展开式的方法，即用几个点上函数值 f 的线性组合来代替 f 的导数，然后按台劳公式展开确定其中的系数，以提高截断误差的阶次。这样既能避免计算 f 的导数，又保证了计算的高精确度。龙格-库塔方法简介如下。

现假定

$$y_{i+1} = y_i + \sum_{m=1}^{v} b_m k_m \tag{3-21}$$

其中 b_m 为待定权重因子，v 为使用 f 函数值的次数，k_m 满足下列方程

$$k_m = hf\left(t_i + c_m h,\ x_i + \sum_{j=1}^{m-1} a_{mj} k_j\right) \tag{3-22}$$

式中 $c_1 = 0,\ m = 1,2,\cdots v$ ，b_m ，c_m 和 a_{mj} 为待定系数。

对于式（3-22）有：

$$\begin{cases} m = 1,\ k_1 = hf\left(t_i, y_i\right) \\ m = 2,\ k_2 = hf\left(t_i + c_2 h, y_i + a_{21}k_1\right) \\ m = 3,\ k_3 = hf\left(t_i + c_3 h, y_i + a_{31}k_1 + a_{32}k_2\right) \end{cases} \tag{3-23}$$

……

其余依此类推。当 v =1 时，只需要一个 k_1，这时式（3-21）就化成前面介绍的欧拉公式。

将 y_{i+1} 展成 h 的幂级数并与微分方程精确解的台劳展开式相比较，便可求得方程（3-22），式（3-23）中的 b_m ，c_m 和 a_{mj} 。

现以 v =2 为例说明龙格-库塔参数的确定方法。

设 v =2，则有

$$k_1 = hf\left(t_i,\ y_i\right) \tag{3-24}$$

$$k_2 = hf\left(t_i + c_2 h,\ y_i + a_{21} k_1\right) \tag{3-25}$$

将 k_1 和 k_2 在同一点（t_i, x_i）上展开，则有

$$k_1 = hf\left(t_i,\ y_i\right) = hf_i \tag{3-26}$$

$$k_2 = hf_i + h^2\left[c_2 \frac{\partial f_i}{\partial t}\bigg|_{t=t_i} + a_{21} f_i \frac{\partial f_i}{\partial y}\bigg|_{y=y_i}\right] + O\left(h^3\right) \tag{3-27}$$

把式（3-26）与式（3-27）代入式（3-21）便得

$$y_{i+1} = y_i + b_1 h f_i + b_2\left[hf_i + h^2\left(c_2 \frac{\partial f_i}{\partial t}\bigg|_{t=t_i} + a_{21} f_i \frac{\partial f_i}{\partial t}\bigg|_{y=y_i}\right)\right] + O\left(h^3\right) \tag{3-28}$$

将 $y\left(t_i + h\right)$ 在 t_i 点按台劳公式展开：

$$y\left(t_i + h\right) = y_i + hf_i + \frac{h^2}{2}\left[\frac{\partial f_i}{\partial t}\bigg|_{t=t_i} + f_i \frac{\partial f_i}{\partial x}\bigg|_{y=y_i}\right] + O\left(h^3\right) \tag{3-29}$$

将式（3-28）与式（3-29）逐项进行比较，令 h 和 h^2 项的系数相等，便可得到下列一组等式：

$$\begin{cases} b_1 + b_2 = 1 \\ b_2\ c_2 = \dfrac{1}{2} \\ b_2 a_{21} = \dfrac{1}{2} \end{cases} \tag{3-30}$$

今有 4 个未知数却只有 3 个方程，故当自由选择一个参数后即可计算出其他参数。

现设 $c_2 = 1$，则

$$a_{21} = 1,\ \ b_1 = b_2 = \frac{1}{2} \tag{3-31}$$

于是就可求得一组计算公式：

$$\begin{cases} k_1 = hf\left(t_i,\ y_i\right) \\ k_2 = hf\left(t_i + h, y_i + k_1\right) \\ y_{i+1} = y_i + \dfrac{h}{2}\left[f\left(t_i, y_i\right) + f\left(t_i + h, y_i + k_1\right)\right] \end{cases} \tag{3-32}$$

由于式（3-32）只取台劳展开式低于 h^2 的各项，式（3-32）叫做为二阶龙格-库塔法，其截断误差为 $O\left(h^3\right)$。

依照上述方法可以推导出三阶和四阶以及四阶以上的龙格-库塔公式。由于四阶及其以下的龙格-库塔公式每次步进一步所需计算函数值 f 的次数等于阶数，而四阶以上的龙格-库塔公式计算函数值 f 的次数比阶数多，其计算函数值的工作量大大增加，而计算精度提高并不快，所以使用得比较少。对于实际工程大量的问题，四阶龙格-库塔公式已可满足精度要求，因此得到了广泛应用。

四阶龙格-库塔公式如下：

$$y_{i+1}=y_i+\frac{h}{6}\left(k_1+2k_2+2k_3+k_4\right) \tag{3-33}$$

式中：

$$\begin{cases}k_1=f\left(t_i,\ y_i\right)\\ k_2=f\left(t_i+\dfrac{h}{2},\ y_i+\dfrac{h}{2}k_1\right)\\ k_3=f\left(t_i+\dfrac{h}{2},\ y_i+\dfrac{h}{2}k_2\right)\\ k_4=f\left(t_i+h,\ y_i+hk_3\right)\end{cases} \tag{3-34}$$

龙格-库塔公式的基本思想是显而易见的：那就是在 $t_i \sim t_{i+1}$ 之间，计算多个点的斜率值（例如二阶龙格-库塔公式为 2 个点，四阶龙格-库塔公式为 4 个点），将多个点斜率值加权平均后用来作为平均斜率的近似值，这样可以得到比欧拉法与梯形法精度更高的计算公式。

3.4.6 算法精度与稳定性比较

下面对上述算法的精度与稳定性作简单比较。

欧拉法计算简单，但精度差。为了提高精度，常常采用减小步长的办法。但是这将导致计算量的增加，有时因舍入误差增大反而使精度降低，所以实际应用很少。

梯形法和预估-校正法具有二阶精度。虽然精度不高，但它们是恒稳格式，对解决某些刚性问题有用。

四阶阿达姆斯法和龙格-库塔法均具有四阶精度。前者计算量较小，但不能自动起动，程序也较复杂；而后者虽然计算量较大，但因为能自动起动且能根据各阶段精度的要求改变步长，其程序编制和使用均较容易，因此得到广泛应用。

对于显式欧拉公式，必须选择足够小的步长 h，才能使之落在稳定域内，所以它是条件稳定的。而隐式欧拉公式是无条件稳定的。除隐式欧拉公式、二阶隐式阿达姆斯法公式为恒稳法外，其他方法都是条件稳定的，步长 h 应限制在系统中最小时间常数的数量级。

另外，当龙格-库塔法的阶数增高时，其稳定域略为增大；而阿达姆斯法的阶数增高时，其稳定域反而缩小。

最后再次着重指出，对于条件稳定的算法，其计算的稳定性与步长有密切关系，只有当步长选择在绝对稳定区域内才能稳定运算。通常，当步长选择得足够小时，一般是可以满足稳定计算要求的。

3.5 SIMULINK 仿真的数值分析

3.5.1 求微分方程数值解的几种算法

在上一节里已经介绍了求解微分方程数值解各种算法的数学原理，并对几种算法的精度与稳定性作了简单比较。

SIMULINK 仿真必然涉及常微分方程组的数值积分。由于动态系统行为的多样性，到现在为止，以上介绍的方法还没有哪一种方法是万能的，能够对所有的模型保证数值仿真结果总是准确、有效而可靠的。为此，MATLAB 的开发者把以上几种求微分方程数值解的方法都提供给 SIMULINK 3.0 仿真工具，并为之编制了不同求数值积分的函数供用户选择使用。

用户须要针对不同类型的模型，按照数值积分法不同的特点、仿真性能与适用范围，正确选择算法并确定适当的仿真参数，以得到最佳仿真效果。

在介绍数值积分算法之前，说明一个重要问题——刚性（Stiff）系统的解算问题。对于一个具有常微分方程组描述模型的系统，如果方程组的 Jacobian 矩阵的特征值相差特别悬殊，此微分方程组则叫做刚性方程组，该系统则称为刚性系统。由于刚性系统的这种特殊性，为了保持数值积分运算的稳定，则使其解算中步长的选取变得特别困难。因此，不是所有的算法而只有那些对于运算稳定性要求不高的算法才能用来解算刚性问题。适合不适合解算刚性问题成为数值积分算法的一个重要性能标识。

以下就介绍 SIMULINK 提供的常微分方程数值解的两大类仿真算法：Variable-step 可变步长类型与 Fixed-step 固定步长类型。

3.5.1.1 Variable-step 可变步长类型算法

这类仿真算法可以让程序修正每次仿真计算的步长大小，并提供错误控制（Error control）以及零点检测（Zero Crossing Detection）的功能。属于 Variable-step 的仿真算法有：oder45、oder23、oder113、oder15s、oder23s 以及 discrete。在这些算法中又可分为“one-step”、“multistep”、“trapezoidal rule”、“discrete”几种，现将各种算法简介如下：

1. “one-step ”

单一步阶算法。这种算法只需要知道前一个时间点的值 $y(t_{n-1})$，就可以立即算出这个时间点的值 $y(t_n)$，也就是说，只需要前一步的解就可以计算出当前一步的解。属于这种算法的有 oder45、oder23、oder23s。

（1）“oder45”：采用显式 Runge-Kutta（4,5）法则与 Dormand prince 组合解法的中阶解法，取代了 SIMULINK 1.x 老版本的“linsim”算法，所以特别适用于仿真线性化程度高的系统。线性系统模型构成的环节大都是由 Transfer Function、State-Space、Zero-Pole、Sun 与 Gain 等标准功能模块组成。为了有合理的输出值，必须要限制最大步长（ Max Step Size）参数的大小。由于 oder45 函数计算快，一般来说，在解题的第一次

仿真时，采用 oder45 是算法的最佳选择。因此，SIMULINK 中的【Simulation】就是以 oder45 为其默认算法的，这种算法应用最广。

（2）“oder23”：采用显式 Runge-Kutta（2,3）、Bogacki 及 Shampine 相组合的低阶解法，用于解决非刚性问题，在容许误差方面以及使用在 stiffness mode （稍带刚性）问题方面，比 oder45 效率高。

（3）“oder23s”：是一种改进的 Rosenbrock 二阶算法。在允许误差比较大的条件下，oder23s 比 oder15s （属于多重步阶算法）更有效。所以在使用 oder15s 处理效果比较差的情况下，宜选用 oder23s 来解决问题。

2. “multistep ”

即多重步阶算法。这种算法需要知道好几个前面时间点的值（即前几步的解）才可以算出这个时间点的值（即当前的解）。属于这种算法的有 oder113、oder15s。

（1）“oder113”：属于 Adams-Bashforth-Moulton PECE 算法，它是计算阶数可以变动（Variable Order）的解法，用于解决非刚性问题。在容许误差要求严格的情况下，比 oder45 函数更有效。 oder113 采用多步法解算。

（2）“oder15s”：属于 numerical differentiation formulas（NDFs）算法，即采用可变阶次的数值微分公式的方法，也称为吉尔（Gear）方法，用于解决刚性（Stiff）问题，当 oder45、oder113 无法解决问题时，可以尝试采用 oder15s 去求解。oder15s 法为多步法，通常运算精度低。

3. “trapezoidal rule”

即梯形法，可以解算刚性问题。

（1）“oder23t”：采用自由内插法实现的梯形法，适用于解决系统有适度刚性并要求无数值衰减的问题。

（2）“oder23tb”：这是用 TR-BDF2 实现的方法，即 Runge-Kutta 公式的第一级采用梯形规则，第二级采用 Gear 法（即二阶 backward differentiation formula）。对于解算误差容限比较宽的问题效果好。适合于解算刚性问题。

4. “discrete” （variable-step）

即用于处理非连续状态的系统模型。

3.5.1.2 Fixed-step 固定步长类型算法

（1）“oder5”：属于 Dormand Prince 算法，就是定步长下的 oder45 解法。

（2）“oder4”：属于四阶的 Runge-Kutta 算法（RK4）。

（3）“oder3”：属于 Bogacki-Shampine 算法，就是定步长下的 oder23 解法。

（4）“oder2”：属于 Heuns's 法则，是欧拉公式（ Euler Formula） 的改进版本。

（5）“oder1”：属于 Euler's 法则。

（6）“discrete（Fixed-Step）”：不含积分运算的定步长方法，适用于解算非连续状态的系统模型问题。

3.5.2 菜单操作方式下仿真算法选择和参数设置

所谓菜单操作方式下的仿真是指：仿真是在 SIMULINK 的模型窗口下，使用其窗口

的主菜单项【Simulation】，运行该项下的【Start】命令即可开始仿真，仿真开始后【Start】变为【Pause】，选中【Pause】可以暂停仿真的执行，要停止仿真可以选择【Stop】。这种仿真方式最直观。在这种仿真方式下，无论是对框图模型本身还是对数值算法的选择及参数的设置都可以很方便地进行操作与修改。

模型不仅在仿真前允许编辑和修改，而且在仿真过程中也允许作一定程度的修改。在菜单操作方式下，允许对模型和框图进行如下实时操作：

（1） 被仿真模块的参数允许有条件地进行实时修改， 其条件是：该参数的变化不改变模型的结构（包括模型几何结构、输入输出维数及状态空间维数）。

（2） 离散模块的采样时间允许实时修改。

（3）允许用游离示波器 （Floating Scope） 实时观察任何一点或几点的动态波形。所谓游离示波器是指在模型视窗里与系统模型没有任何可见连接线的示波器。

（4）在一个系统仿真的同时，允许打开另一个系统。

在仿真之前通常要对仿真参数进行设置，这就是主菜单项【Simulation】下【Parameters】的菜单命令要完成的任务。对于比较简单的模型，可以使用系统的缺省值而不用进行过多的参数设置。设置仿真参数是采用菜单方式进行仿真最主要的工作。

3.5.2.1 解算器（Solver）标签页的参数设置

在仿真模型的SIMULINK 窗口主菜单中，执行菜单【Simulation】下的【Parameters】菜单命令后，SIMULINK会弹出仿真参数设置对话框标签之一为“Solver”解算器标签页，如图 3-26 所示。图中选择的算法是“Variable-step ”可变步长类型。如果选择“Fixed-step”固定步长类型的算法，标签页“Solver”如图 3-27 所示。

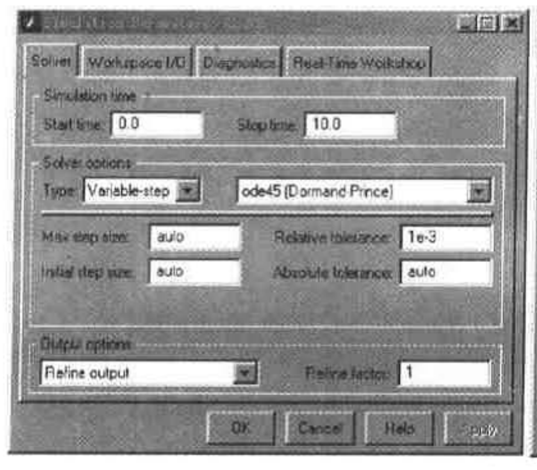


图 3-26 解算器可变步长仿真参数设置窗口

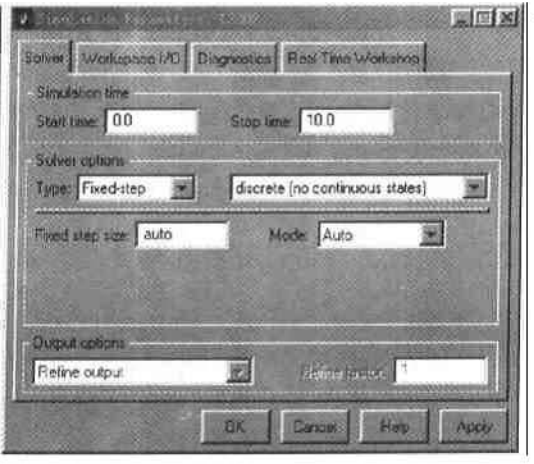


图 3-27 解算器固定步长仿真参数设定窗口

解算器（Solver）标签页参数设定是进行仿真工作前准备的必须步骤，如何设定参数是根据要解决问题的要求而决定的，以便使 SIMULINK 发挥最佳的仿真效果。最基本的参数设定包括仿真的起始时间与终止时间、仿真的步长大小与解算问题的算法等等。进行参数设定可以直接选择主菜单项【Simulation】下的【Parameters】命令，出现“Solver”

对话框后进行设定。图 3-26 的解算器（Solver）标签页参数设定窗口中选项的意义如下：

（1）“Simulation time” 栏为设置仿真时间，在“Start time”与“Stop time”旁的编辑框内分别输入仿真的起始时间与停止时间，其单位是“秒”。系统实际运行时间与设置输入的时间“秒”数不会一致，因为实际运行时间与计算机的性能、模型复杂程度、解题所选择的算法及步长、要解决问题的误差要求等诸多因素有关。

（2）“Solver options”栏为选择算法的操作，包括许多选项。“type”栏的下拉式选择框中可选择可变步长（Variable-step）算法 （界面如图 3-26 所示） 或者固定步长“Fixed-step”算法（界面如图 3-27 所示）。

“Variable-step”这种方式能够在仿真过程中自动修改步长的大小（Step Sizes）以满足容许误差设定与零交叉 （Zero Crossing）检验的需求（设置零交叉检验可以提高仿真精度，但对仿真速度有所影响）。属于这种方式的有 orer45、oder23、oder113、oder15s、oder23s、oder23t、oder23tb、discrete 多种方法可供选择。一般系统设定 oder45 方法为默认解题算法，但是对离散系统模型时，就必须选用 discrete 算法。

“Max step size”栏为设定解算器运算步长的时间上限，“Initial step size” 为设定的解算器第一步运算的时间，一般默认值为“auto”。相对误差“Relative tolerance”的默认值为 1e-3，绝对误差 “Absolute tolerance”的默认值为“auto”。

“Fixed-step”（图 3-27）这种方式能够固定步长的大小不变，不自动修正步长的大小以满足容许误差的设定与零交叉的需求。其显示的内容与 Variable-step 的不同。其解题算法共有 oder5,oder4,oder3,oder2,oder1,discrete 几种可供选择，一般采用 oder4 作为解题算法，它等效于 oder45。另外 oder3 等效于 oder23。固定步长方式只可以设定 fixed step size 为“自动”。这种方式下虽然没有 output options 的设定问题，却增加了一选项“Mode”栏以选择模型的类型。该栏有三个选项：“MultiTasking”（多任务）、“SingleTasking”（单一任务）与“auto”（自动）。“MultiTasking”模型指其中有些模块具有不同的采样速率，并对模块之间采样速率的传递进行检测；“SingleTasking”模型各模块的采样速率相同，不检测采样速率的传递；“auto”则根据模型中模块采样速率是否相同，决定采用前两者的哪一种。

（3）“output options” 输出选择栏的第一选项为细化输出（ Refine Output），其细化系数（ Refine Factor）最大值为 4，缺省值为 1，数值愈大则输出愈平滑。

第二选项为产生附加输出 （roduce additional output），允许指定产生输出的附加时间（Output Times）。该项被选中后，在编辑框“Output times”中可以输入产生输出的附加时间。这种方式可改变仿真步长以使其与指定的附加输出时间相一致。

第三个选项为只产生特定的输出 （Produce Specified Output Only），只在指定的输出时间中产生仿真输出。这种方式可改变仿真步长以使其与产生输出的指定时间相一致。

（4）在标签页的右下部有四个按钮，它们的功用分述如下。

【Ok】按钮用于参数设置完毕，可将窗口内的参数值应用于系统以下的仿真，并关闭对话框。

【Cancel】按钮用于立即撤销参数的修改，恢复标签页原来的参数设置，关闭对话框。

【Help】按钮用于打开并显示该模块使用方法说明的帮助文件。

【Apply】按钮用于修改参数后的确认，即表示将目前窗口改变的参数设定应用于系统以下的仿真，并保持对话框窗口的开启状态，以便进一步修改。

这种 4 个按钮的组合，在其他许多界面里都有，其功能与此相同。

3.5.2.2 工作空间（Workspace I/O）标签页参数设置

仿真控制参数 Simulation Parameters 设定对话框标签之二为“Workspace I/O”工作空间标签页，如图 3-28 所示。

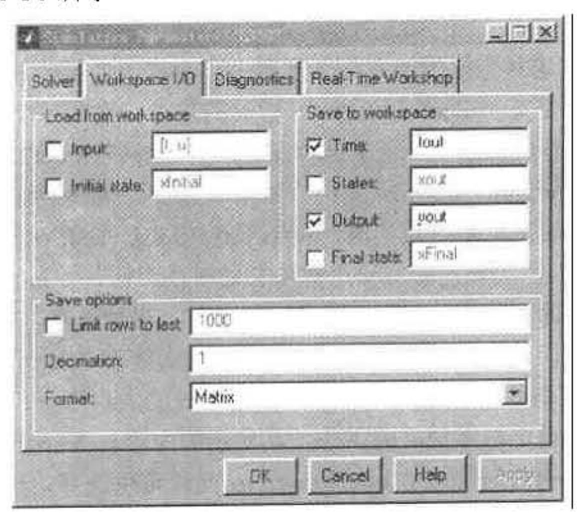


图 3-28　设定工作空间 Workspace 参数窗口

在这一标签页中设置参数后，可以从当前工作空间输入数据、初始化状态模块（State）、把仿真结果保存到当前工作空间。

（1）“Load from workspace”栏可以从 MATLAB 工作空间获取数据输入到模型的输入模块（In1），这是 SIMULINK 的一个重要功能。当然，模型一定要有输入模块（In1）。具体操作方法是：勾选“Input”栏，在其后编辑框里输入数据的变量名，变量名缺省值为[t,u]，t 是一维时间列向量，u 是与 t 相同的二维列向量。如果输入模块有 n 个，则 u 的第 1、2、…、n 列分别送往输入模块 In1、In2、…、Inn。

（2）勾选“initial state”栏，将迫使模型从 MATLAB 工作空间获取模型中全部模块所有状态变量的初始值，这就是初始化状态模块（State）。全部模块当然包括积分模块，不论它原来的情况如何均应同样初始化。该栏后的编辑框里填写的含有初始值变量（其缺省名为 xInitial）的个数应与状态模块数目一致。

（3）把仿真结果保存到 MATLAB 当前工作空间，此项功能在“Save to workspace”栏中设置，可以设定 Time、State、Output、Final state 诸项。

勾选“Time”栏，模型把变量（Time）以指定的变量名（缺省名为 tout）存储在 MATLAB 工作空间。

勾选“State”栏，模型把状态变量以指定的变量名（缺省名为 xout）存储在 MATLAB 工作空间。

勾选“Output”栏，对应着模型窗口中使用的输出模块“out1”，须在 MATLAB 工

作空间填入输出数据变量名（输出矩阵缺省名为 yout），输出矩阵每一列对应于模型的多个输出模块“out”，每一行对应于一个确定时刻的输出。

勾选“Final state”栏，模型把状态变量的最终状态值以指定的名称（缺省名为 xFinal）存储在 MATLAB 工作空间。状态变量的最终状态值还可以被模型再次调用。

（4）“Save options”变量存储选项栏须与“Save to workspace”栏配合使用。

勾选“Limit rows to last”栏，选择框可以限定可存取的行数。

“Decimation”栏为设置的降频程度系数，降频系数的默认值为 1，表示每一个点都返回状态与输出值；若设定为 2，则会每隔断 2 个点返回状态与输出值，这些结果会被保存起来。

“Format”栏的下拉式选择框中，提供了三种保存数据的格式选择：矩阵（Matrix）、构架（Structure）、带时间的构架（Structure with time）。

在 Workspace I/O 标签页下的四个按钮，其操作功能同解算器（Solver）标签页内说明。

3.5.2.3 诊断（Diagnostics）标签页参数设置

仿真控制参数设定对话框标签之三为“Diagnostics”诊断标签页，如图 3-29 所示。

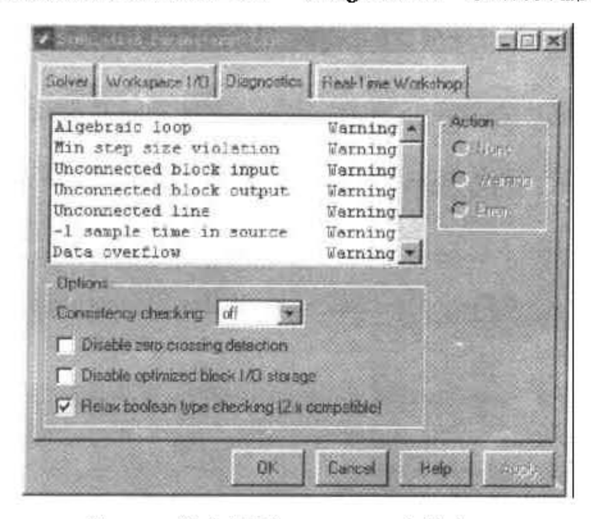


图 3-29 设定诊断 Diagnostics 参数窗口

诊断 Diagnostics 标签页用于诊断模型是否精确，是否出现异常情况，效果是否良好，或者在发生某些事件时，设定应采取的措施与作出的反应。在诊断 Diagnostics 标签页中部空白编辑框内显示程序执行时可能遇到的情况，而“Action”栏为异常情况发生时应作出的反应，这些都是由用户选择设定的。而“Options”为异常情况发生时应执行的操作。

“Action”栏作出的反应有 3 类：【None】（不做反应），【warning】（警告），【error】（提示错误）三种单选钮选择。需要说明，警告信息出现时不影响程序的运行，而提示错误出现时程序要停止运行，需要采取相应的措施。

“Consistency Checking”栏用于对模型是否进行一致性检验。一致性检验用以验证

微分方程解算中的某些假设，以保证操作 s 函数与 SIMULINK 内置模块具有相同的规则；还确保模块对于固定的时间 t 输出是不变的。一致性检验栏后的下拉式选择框有三种选择：off、warning、error。对于解算器（Solver）设置后的一致性检验，一般都选“off”，以免影响解算速度。如果选择 “warning” 或“error”，SIMULINK 会验证输出值、零交叉（Zero Crossing）、微分值与状态值是否正确等。

“ Disable zero crossing detection” 复选框被勾选时，则取消对零交叉的检测，但是如果模型内有 Hit Crossing 功能模块时，最好不要选该复选框。

“ Disable optizmized block I/O storage”栏用来选择对模块的 I/O 值能否优化使用内存缓冲区。该栏被勾选时，则每个模块的 I/O 值都分配到单独的内存缓冲区，这样会增加内存的使用量，所以一般只有调试模型时才选。

“ Relax boolean type checking（2.x compatible）” 栏用于是否放宽对于逻辑类型数据的检查，以使 Simulink3.0 与 Simulink 2.x 兼容。该栏被选中时，系统允许对 double 类型数据进行逻辑操作，即把 double 类型数据作为逻辑类型数据来处理。

在 Diagnostics 标签页下部的四个按钮，其功能也同解题算法（Solver）标签页的介绍。

3.5.2.4 实时工作空间（RTW）标签页参数设定

仿真参数设定对话框标签之四为：“ Real-Time Workshop（RTW）” 实时工作空间标签页， 如图 3-30 所示。

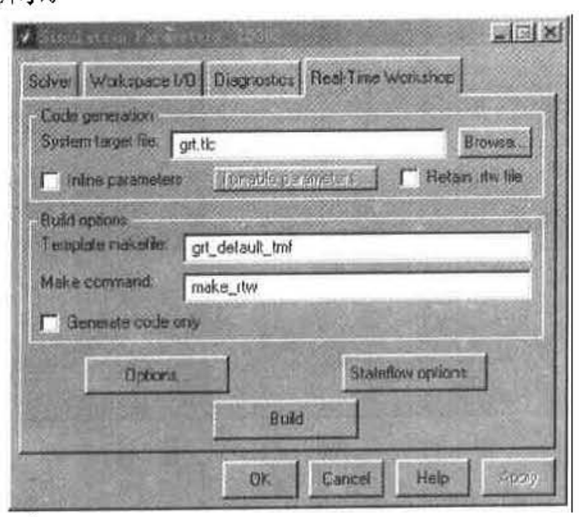


图 3-30 实时工作空间（RTW）标签页参数设定

实时工作空间 Real-Time Workshop（RTW）是 Simulink 的一个重要功能模块，它也是一种实时开发环境，在该环境下可直接从 Simulink 模型生成可移植的程序源代码（C 语言与 Ada 语言代码），并自动生成能在多种环境中（包括实时系统与单机仿真）实时执行的程序。RTW 主要用来进行实时控制、实时仿真、交互式实时参数调整与生成可移植的 C 语言代码等。

在 Simulink 模型窗口里打开【Simulation】菜单下的【Parameters】命令对话框，选择 RTW （Real-Time Workshop） 标签页选项就进入了如图 3-30 所示的实时工作空间

（RTW）标签页参数设置对话框。还可以在模型窗口里打开【Tools】菜单，使用【RTW Options】命令，同样也会打开 Real-Time Workshop 标签页设置对话框。

在“Code Generaton”代码生成栏下，“System Target File”栏用来指定系统目标文件的代码及其类型，并可以用【Browse】按钮打开进行查看。

“Inline parameters”内联参数栏是用来从运行状态的模型中对采样时间为常数的模块对其选择移动的方式。当该栏被选中时，Simulink 自动把所有内联参数设置为常量，并激活【Tunable parameters】可调参数按钮。【Retian rtw file】栏被勾选时为保留 rtw 文件。

“Build options”构造选项栏下，“Template Makefile”模板文件（扩展名为为.tmf）只有在“Generate code only ”仅生成代码栏的复选框不选中时才使用。RTW 把模板文件的代码编译为可以执行文件，“Generate code only”栏仅用来生成代码。

“Make command”栏是 make 文件宏指令 MAKECMD 定义的 make 命令，它在 RTW 把模板文件的源代码编译为可以执行文件时使用。

RTW 标签页下有三个按钮：【Options】、【Stateflow options】、【Build】。

【Options】按钮按下后，打开代码生成选项“Code Generation Options”对话框。

【Stateflow options】为状态流程图选项按钮。

【Build】按钮按下时，RTW 自动调用 make_rtw 控制开始构造过程：先由 make_rtw 编译模型的模块框图，生成 model.rtw 文件，尔后 make_rtw 启动目标语言编译器生成目标文件代码。

3.5.3 在 MATLAB 命令窗口中的指令工作方式

使用菜单命令方式与在 MATLAB 命令窗口中的指令工作方式，这两种进行仿真的方法各有所长，互相补充。

3.5.3.1 两种仿真方法的使用

已经介绍的使用菜单命令进行仿真的方法，有以下优点：

（1） 可以直接设置及修改许多仿真参数与模块参数，包括仿真时间、解题算法、仿真步长等等。

（2） 可以同时对几个系统进行仿真。

（3） 可以实时监视连线上传输的数据。

（4） 菜单命令仿真方法是用鼠标操作的方法，只需用鼠标点击菜单即可完成仿真操作任务，确实简单、方便、快捷。

将要介绍的另外一种方法是在 MATLAB 命令窗口中输入命令进行仿真。这种方法虽然不比菜单命令仿真方法简单、方便、直观，但是它的特长优势也是菜单命令不能替代的。

这种方法具体如下：

（1） 可以用 M 文件进行仿真，仿真参数与模块参数也可反复修改，或者在设定的条件下（可重新设定模块的初始值及指定使用比较复杂的外部输入函数）开始仿真或停止仿真。

（2） 可以仿真由 M 文件或者 MEX 文件（用 C 或 Fortran 编写的可以被 MATLAB

调用或执行的动态连接子程序）描述的系统动态模型。

（3）如果在调用函数时不指定输出变量，则会自动绘制系统的状态轨迹，或者在有输出模块时绘制系统的输出。

（4）这种方法仿真的速度比菜单命令仿真方法的快。

由上可见，命令窗口中指令工作方式的优点也不容忽视。

3.5.3.2 指令工作方式仿真使用的命令

在 MATLAB 命令窗口中输入命令进行仿真所使用的函数命令有：sim（）、simset（）、simget（）、set_param（）。

1．运行 SIMULINK 模型的 sim（）函数命令

命令调用的格式是：

[*t*, *x*, *y*] = sim（*'modelname'*, *timespan*, *options*, *ut*）

其中：

（1）在这调用格式中，除第一输入参数“*modelname*”是必需的之外，其余输入参数缺省（即没有赋值）时，系统会自动以【Simulink Parameters】中设置的参数值作为仿真设置。

（2）输入参数“*modelname*”是方框图模型的（M 或 MEX）文件名，为字符串变量，必须用单引号括起来，这是在任何一种调用格式中必不可少的部分。

（3）第二个输入参数“*timespan*”是仿真时间信息。有三种格式：

[tFinal] 指定仿真停止时间，仿真开始时间缺省值为 0；

[tStart tFinal] 指定仿真开始时间与停止时间；

[tStart OutputTimes tFinal] 指定仿真开始时间、要输出的时间及停止时间点。

（4）第三个输入参数“*options*”是由命令“simset”设置的仿真参数。

（5）第四个输入参数“*ut*”是被仿真系统外部输入到输入模块的数据。

（6）输出参数的含义为：

t 仿真（即取积分值）的时间向量；

x 系统状态模块的状态矩阵；

y 系统的输出矩阵，每一列对应一个输出端口（Output）的值。

【例 3-3】试对图 3-31 所示的正弦波信号经积分与微分后的信号，在 MATLAB 命令窗口中输入命令进行仿真。

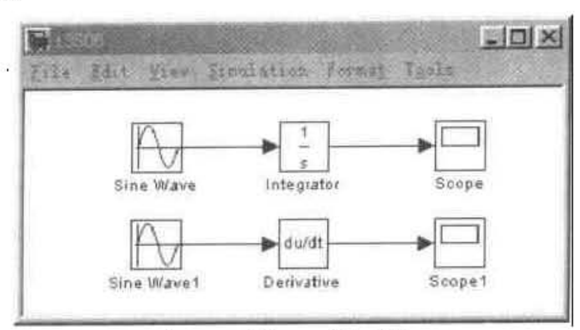


图 3-31 例 3-3 系统模型

【解】

（1）sim（）函数命令只带必不可少的参数

在 MATLAB 命令窗中执行以下命令，在结构图模型 t3506.mdl 里鼠标左键点击两个示波器可得图 3-32 所示的仿真曲线，左图为积分输出，右图为微分输出。

```
[t,x,y]=sim('t3506');
```

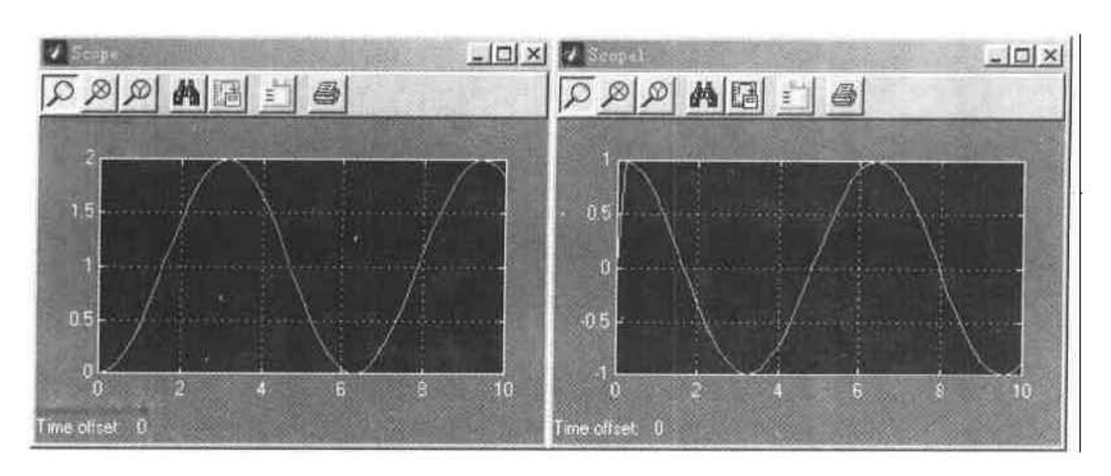


图 3-32　sim（）命令只带必要参数的积分与微分环节仿真曲线

（2）sim（）命令指定仿真开始与停止时间的参数

在 MATLAB 命令窗中执行以下命令：

```
[t,x,y]=sim('t3506',[1,8]);
```

在结构图模型 t3506.mdl 里鼠标左键点击两个示波器可得图 3-33 所示的仿真曲线，左图为积分输出，右图为微分输出。

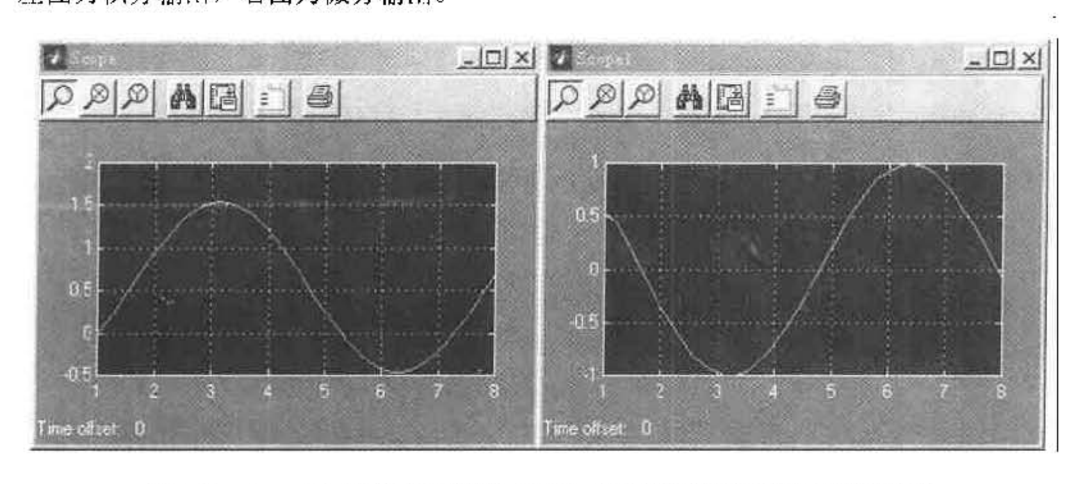


图 3-33　sim（）命令指定仿真开始与停止时间的积分与微分环节仿真曲线

（3）sim（）命令指定仿真时间点的参数

在 MATLAB 命令窗中执行以下命令：

```
[t,x,y]=sim('t3506',[2,4,6,8]);
```

在结构图模型 t3506.mdl 里鼠标左键点击两个示波器可得图 3-34 仿真曲线，左图为积分输出，右图为微分输出。

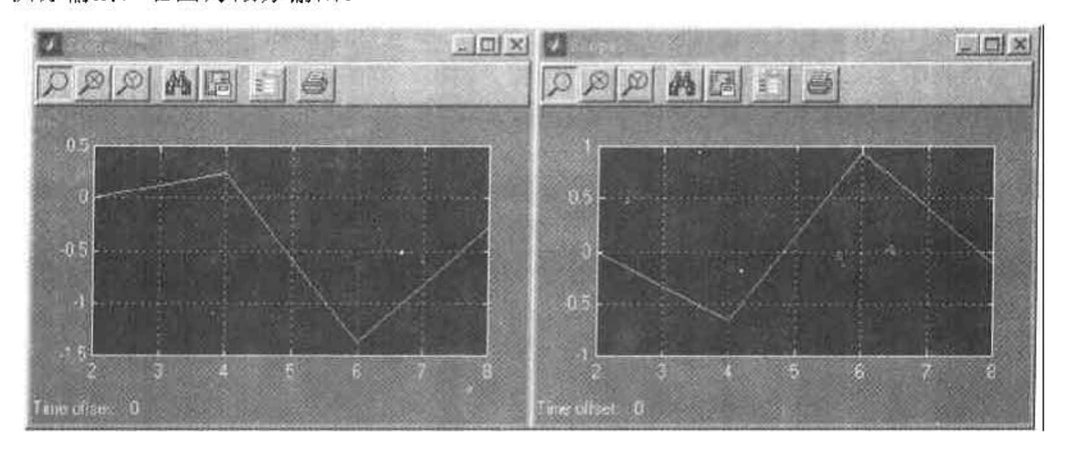


图 3-34 sim（）命令指定仿真时间点的仿真曲线

2. 设置仿真参数的 simset（） 函数命令

simset（） 函数命令用来为 sim（） 函数命令设置或编辑仿真参数，并把设置结果保存在一个特殊的结构变量中。simset（） 函数命令的调用格式有四种：

（1）*options* = **simset** （*property,value*，…）；

这种格式是把“*property*”的参数设置为“*value*”，结果保存在结构变量 *options* 中。

（2）*options* = **simset** （*oldopstruct*，*property*，*value*，…）；

这种格式是把已经存在的结构“*oldopstruct*”（由 *simset* 产生）的参数“*property*”重新设置为“*value*”，结果保存在新结构变量 *options* 中。

（3）*options*=**simset**（*oldopstruct*，*newopstruct*）；

这种格式是用结构“*new_opstruct*”的值替代已经存在的结构“*old_opstruct*”的值。

（4）**simset**（）不带参数。

这种格式是显示所有的参数名与它们可能的值。

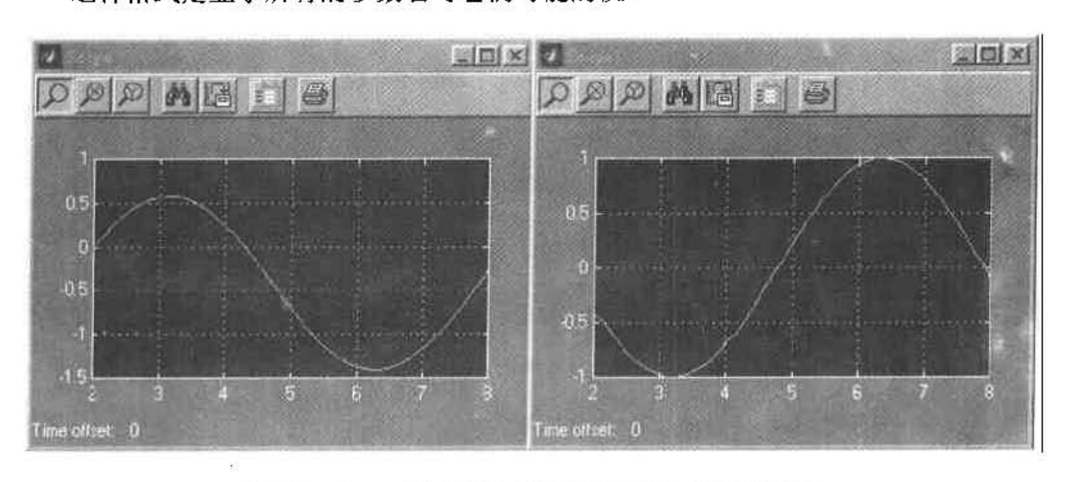


图 3-35 simset（）函数命令指定仿真时间点的仿真曲线

【例 3-4】（续【例 3-3】），试在 MATLAB 命令窗口中用 simset（）函数命令进行仿真。

【解】

根据 simset（）函数命令的调用格式（1）与 sim（）函数命令调用格式的说明（3）、（4），给出以下指令：

```
option1=simset('OutputVariables','xy','OutputPoints','all');
[t,x,y]=sim('t3506',[2,4,6,8],option1);
```

指令运行后，在结构图模型 t3506.mdl 里鼠标左键点击两个示波器，产生如图 3-35 所示的仿真曲线，左图为积分输出，右图为微分输出。

由仿真指令可以看出，仿真设置的输出变量为 *x*，*y*,输出的点数设置为全部的点。所以图 3-35 所示的仿真曲线比 图 3-34 的仿真曲线要光滑了很多。

3. 获取模型仿真参数的 simget（）函数命令

simget（）函数命令用来获得模型仿真参数的设置值。simget（）函数命令的调用格式有三种：

（1）*struct*=**simget**（*modelname*）；

这种格式是返回指定模型 *modelname* 的参数设置的 *options* 结构。

（2）*value*=**simget**（*modelname*，*property*）；

这种格式是返回指定模型 *modelname* 的参数 *property* 的值。

（3）*value*=**simget**（*OptinonStructure*，*property*）；

这种格式是获取参数设置 *options* 结构 *OptionStructure* 中的参数 *property* 的值。

【例 3-5】（续【例 3-4】），试在 MATLAB 命令窗口中用 simget（）函数命令获取模型仿真参数。

【解】

给出以下调用 simget（）函数的指令：

```
struct=simget('t3501')
```

指令执行后可得模型仿真参数为：

```
struct =

          AbsTol: 1.0000e-006
           Debug: 'off'
      Decimation: 1
    DstWorkspace: 'current'
  FinalStateName: ''
       FixedStep: 'auto'
    InitialState: []
     InitialStep: 'auto'
        MaxOrder: 5
      SaveFormat: 'Matrix'
         MaxRows: 0
         MaxStep: 'auto'
```

```
    OutputPoints: 'all'
 OutputVariables: 'ty'
          Refine: 1
          RelTol: 0.0010
          Solver: 'ode45'
    SrcWorkspace: 'base'
    Trace: ''
       ZeroCross: 'on'
```

4．设置仿真参数的 set-param （ ）函数命令

这种格式是用 set-param （ ）函数命令来设置 Simulink 仿真参数，控制仿真开始、暂停、终止进程。set-param（ ）函数命令调用格式有设置仿真参数与控制仿真进程两种。

（1）**set-param**（*modelname,property,value,...*）；

命令格式中 *modelname* 为设置的模型名，*property* 为要设置的参数，*value* 是设置值。这里设置的参数可以有很多个，而且和用 simset 设置的内容不尽相同。

（2）**set-param**（*modelname,'SimulationCommand','cmd'*）

命令格式中 *modelname* 为模型名，而 *'cmd'* 就是控制仿真进程的各个命令，包括：'start'，'stop'，'pausue'，'continue' 或者'update'。

在用这两个命令的时候，需要注意必须先把模型打开。

【例 3-6】(续【例 3-3】)，试在 MATLAB 命令窗口中用 set-param （ ）函数命令进行仿真。

【解】

根据 set-param （ ）函数命令的调用格式（1）与（2），给出以下 MATLAB 指令：

```
set_param('L3501','StartTime','10','StopTime','30')
set_param('L3501','SimulationCommand','Start')
```

指令运行后有如图 3-36 所示的仿真曲线，左图为积分输出，右图为微分输出。

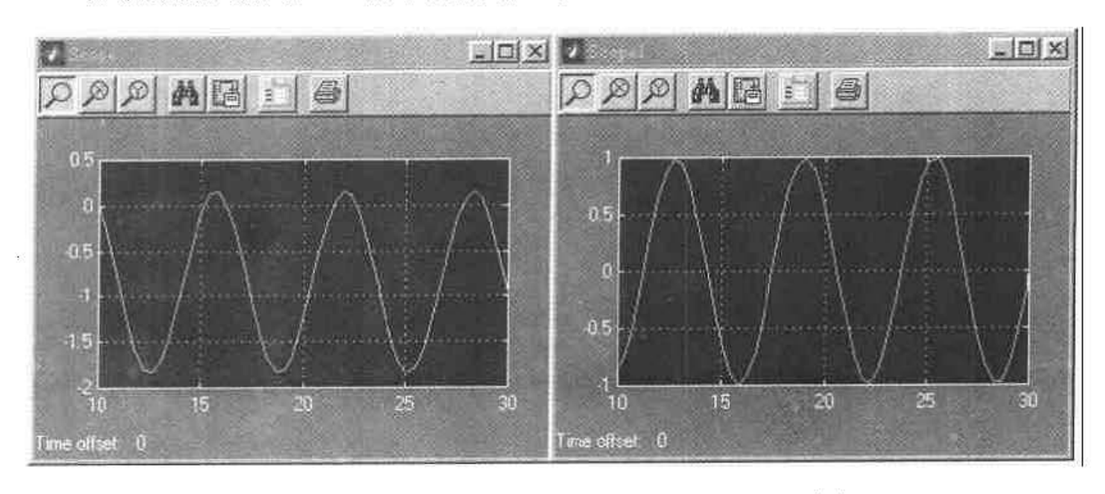


图 3-36　set-param （ ）函数命令指定仿真时间点的仿真曲线

3.6 观察 SIMULINK 的仿真结果

控制系统仿真后的结果，可以用 SIMULINK 提供的许多观察工具加以查看，还可以利用 SIMULINK 提供的分析工具对仿真结果进行分析。

SIMULINK 的 Sinks 输出模块库中的几个模块都可以用来观察仿真结果。

第一种模块是将仿真结果信号输入到输出模块“Scope”示波器、“XY Graph ”二维 X-Y 图形显示器与“Display ”数字显示器中直接查看图形或者数据。

第二种是将仿真结果信号输入到“To Workspace”模块中，即保存到 MATLAB 工作空间里，再用绘图命令在 MATLAB 命令窗口里绘制出图形。

第三种是将仿真结果信号返回到 MATLAB 命令窗口里，再利用绘图命令绘制出图形。

3.6.1 使用示波器模块观察仿真输出

已经介绍过，在 Simulink 库浏览器的“Sinks”输出模块库（图 3-10）中，有“Scope”、“XY Graph” 与“Display”三个示波器模块，是用来观察仿真输出的。

（1）“Scope”：将信号显示在类似示波器的图标窗口内，可以放大、缩小窗口，也可以打印仿真结果的波形曲线。

（2）“XY Graph”：绘制 X-Y 二维的曲线图形，两个坐标刻度范围可以设置。

（3）“Display”：将仿真结果的信息数据以数字形式显示出来。双击“Display”图标，打开其“Block Parameters:Display”对话框。其中“Format”栏设置数字显示格式；“Decimation”栏设置数据显示的抽选频度：1 为缺省设置，表示每点都显示，n 为隔（n-1）点显示一次；“Sample time”栏设置显示时间间隔，-1 为忽略采样时间间隔。

只要将这三种示波器图标放在控制系统模型结构图的输出端上，就可以在系统仿真时，同时看到仿真输出结果。“Display” 会将结果数据直接显示在模块的窗口中（图 3-37），“Scope”及“XY Graph”会产生新的窗口（图 3-38、图 3-39）。

【例 3-7】用三种示波器观察正弦波信号的波形曲线（图 3-37）。

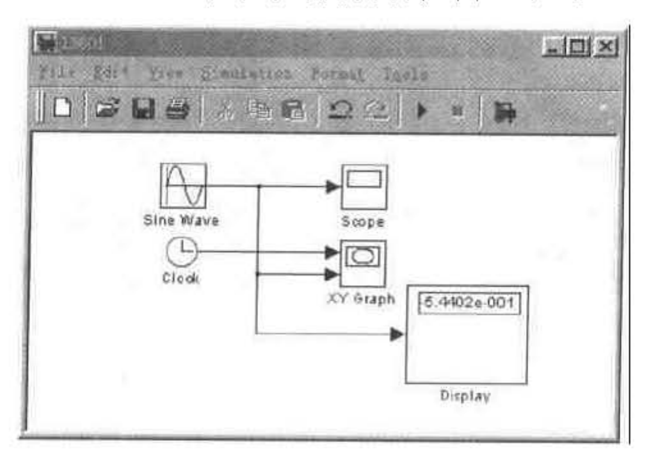


图 3-37 三种示波器观察正弦波

在图 3-37 主菜单选择【Simulation】中的【Start】子菜单项命令，进行系统仿真。仿真结束后，可以在“Scope”与“XY Graph”中分别看到正弦波信号的波形曲线，如图 3-38 与图 3-39 所示，这两图波形完全相同。在“Display”模块图标的小窗口里直接显示仿真数值信息（图 3-37）。

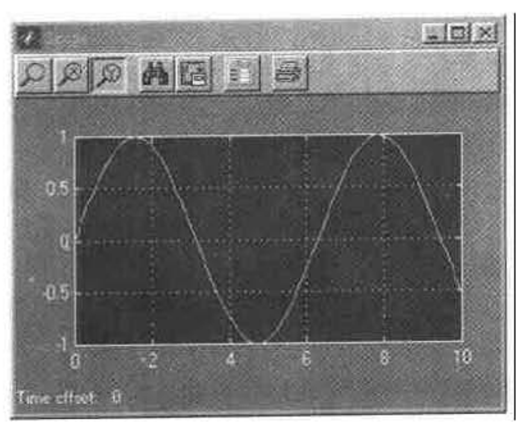


图 3-38 示波器中的正弦波形

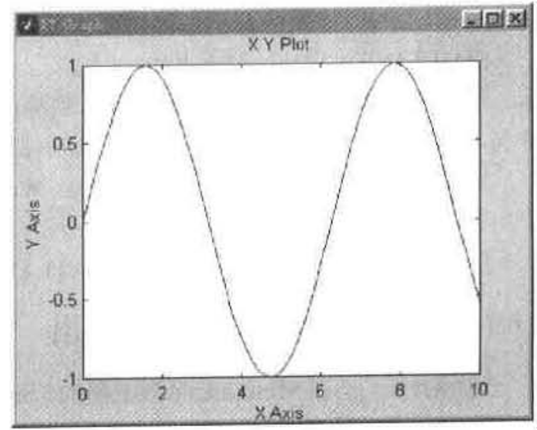
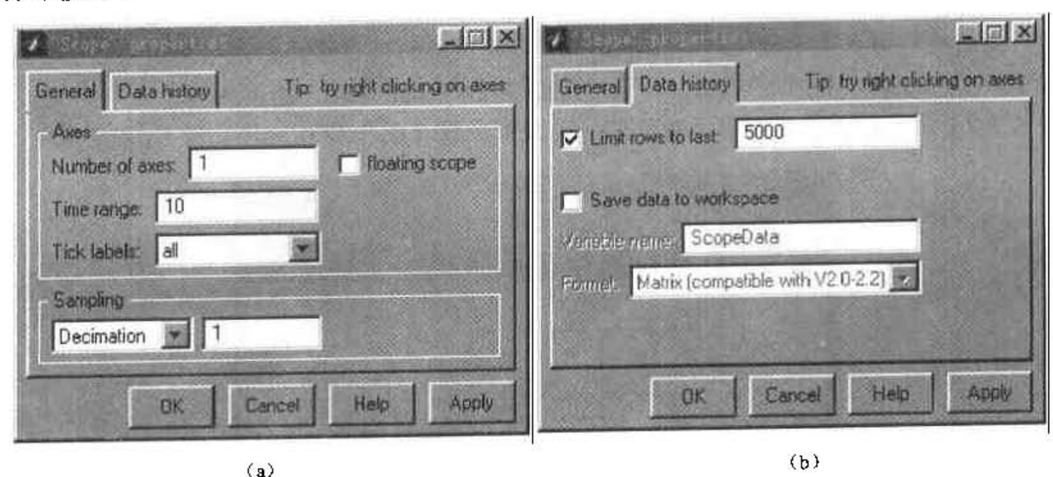


图 3-39 二维图形中的正弦波形

由于示波器使用极其普遍，有必要将示波器的使用方法稍作介绍。

由图 3-38 看到，示波器窗口的标题是“Scope”，标题栏下是工具栏。左边的三个按钮“■■■”分别为 X-Y 双向变焦（Zoom）管理、X 轴向变焦（Zoom X）管理、Y 轴向变焦（Zoom Y）管理；“■”为对于示波器窗口纵坐标刻度的自动管理；“■”为保存当前轴的设置；“■”打开示波器属性对话框（图 3-40）；工具栏最右边的是打印按钮。



(a)　　　　(b)

图 3-40 示波器属性对话框

示波器属性对话框中有两张标签页，图 3-40（a）为一般参数设置，图 3-40（b）为数据存储参数设置。

图 3-40(a)中的参数设置主要是针对示波器窗口的坐标系与曲线显示方面的设置。

“Axes”栏下的“Number of axes”为示波器窗口内的坐标系个数，缺省设置为 1；当设置为 2 时，相应模型结构图中示波器图标的输入端也变为两个输入端口。其中“Time range”栏为信号显示从 0 开始的时间区间，缺省设置为 10，若设置 n，则信号显的时间区间为[0,n]；“Tick labels”下拉三个选项：“all”为坐标系标注标识“Time offset 0”，“none”为坐标系不标注标识“Time offset 0”，“bottom axis only”为坐标系底部标注标识“Time offset 0”，实际上与“all”选项相同。“floating scope”栏被勾选时，则示波器为游离状态，模型结构图中示波器图标的输入端将与系统模型的连线会断开。

“Sampling”下拉菜单有两个选项：其一，Decimation 设置数据的显示频度：1 为缺省设置，表示每点都显示；设置为 n 时（在下拉菜单框右的空白编辑框内输入），则为隔（n-1）点显示一次。其二，Sample time 设置显示点的采样时间间隔，缺省设置为 0，意为显示连续信号；设置-1 时为显示方式由输入信号决定。

图 3-40（b）的参数设置，主要是针对示波器的数据存储与传送方面的。

“Limit rows to last ”栏设置缓冲区存储数据的长度，缺省设置为 5000。若输入的数据过多时，则会自动清除原有的数据。

“Save data to workspace”栏用来把示波器缓冲区存储的数据送到 MATLAB 工作空间。缺省设置为不被勾选，意为不送数据到 MATLAB 工作空间。

“Variable name”是存储数据的变量名，可以设置，也可以用缺省设置名“ScopeData”。

“Format”为三种保存数据的格式选择：Matrix（矩阵）、Structure（构架）、Structure with time（带时间的构架）。

3.6.2 使用 To Workspace 模块将仿真输出信息返回到 MATLAB 命令窗口

如果不用示波器直接观察仿真结果，可以将控制系统仿真结果信号输入到“To Workspace”（MATLAB 的工作空间）模块中。该模块会自动将数据输出到 MATLAB 命令窗口里，并用变量保存起来，再用绘图命令在 MATLAB 命令窗口里绘制出图形。

【例 3-8】利用“To Workspace”模块传送数据到 MATLAB 命令窗口并绘制正弦波形曲线（图 3-41）。

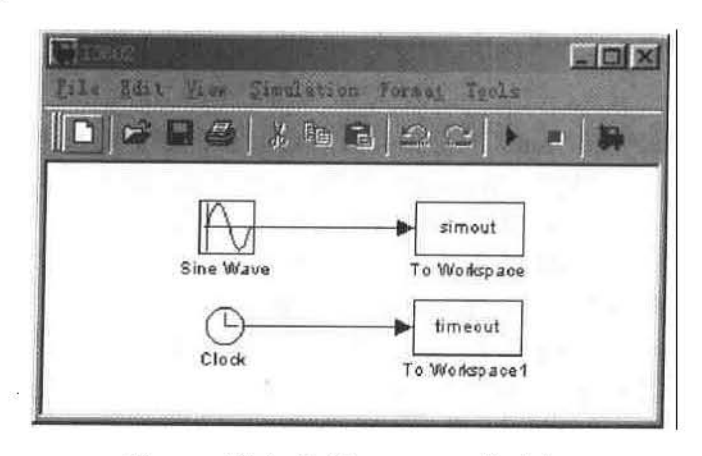


图 3-41　利用工作空间 Workspace 传送数据

正弦波输出到“To Workspace”模块，该模块会将信息、数据返回到MATLAB命令窗口中，并用一个名为“ simout ”的变量保存起来（就是图 3-41 中控制模块上写的 “simout” ）。只要执行图 3-40 中主菜单项【Simulation】下的【Start】命令，MATLAB 就会自动将每个时间数据存入 MATLAB 命令窗口中，用“tout”这个变量保存起来。

此外，可以用输入源“Sources”模块库中的“Clock ”模块来查看“Workspace” 输出的时间数据（图 3-41）。这只要将 “Clock” 模块输出到 “To Workspace1” 模块中，并将“To Workspace1”的输出变量改名。执行的操作为：鼠标左键双击“To Workspace1”模块，打开其对话框（图 3-42），修改“Variale name” ，将变量名改为 “timeout” 后，点击按钮【OK】即可。图 3-42 To Workspace 模块对话框中，“Maximum number of rows”栏用来限定存储的最多数据点数，若送入数据过多则自动清除旧数据，若设为“inf”则保存全部数据。Decimation 栏设置传送数据的频度，1 为缺省设置，表示每点都传送：n 为隔（n-1）点传送一次。Sample time 栏设置传送时间的间隔，-1 为忽略采样间隔。Save format 栏为存储数据的三种可选格式：Matrix（矩阵）、Structure （构架）、Structure with time （带时间的构架）。

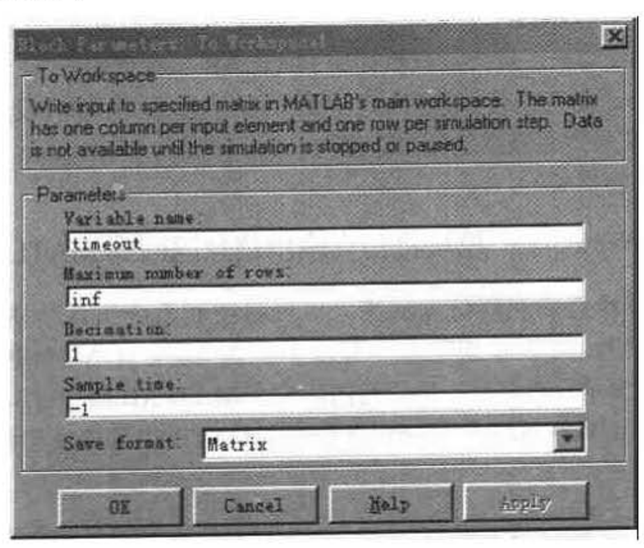


图 3-42　To Workspace 模块对话框

可以验证系统经变量“tout”自动传到 MATLAB 命令窗口里的时间信息和“Clock”与“To Workspace1” 模块经 “timeout” 变量保存的时间信息是否一致。要进行这个验证，须做如下操作：回到 MATLAB 命令窗口，输入命令 who 查询在 MATLAB 命令窗口里有多少个变量。执行 who 命令后，发现有 “simout”、“timeout”、“tout” 三个“out”变量.。当输入指令“out=[tout timeout]”来查看变量的内容时，指令执行结果显示“tout”与“timeout”保存的时间信息完全相同（图 3-43）。

控制系统输出数据与时间数据返回到 MATLAB 命令窗口之后，还可以用绘图命令在 MATLAB 命令窗口里绘制出图形。指令如下：

```
plot (tout,simout)
```

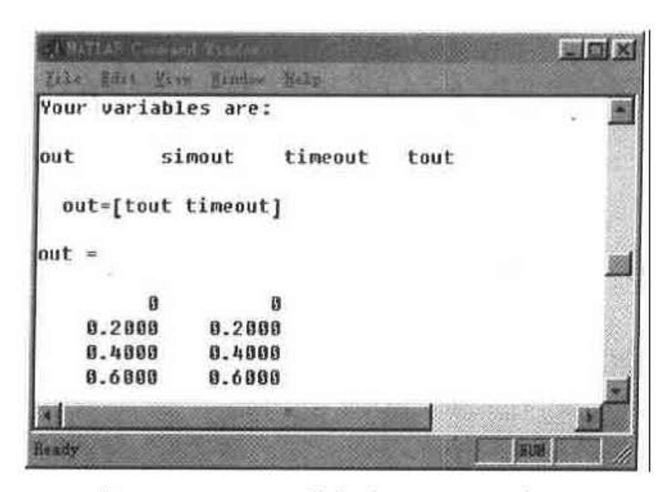


图 3-43　Workspace 输出到 MATLAB 的变量

指令执行后绘制的正弦曲线如图 3-44 所示，可以看出，图 3-44 与图 3-38 或图 3-39 是一样的。

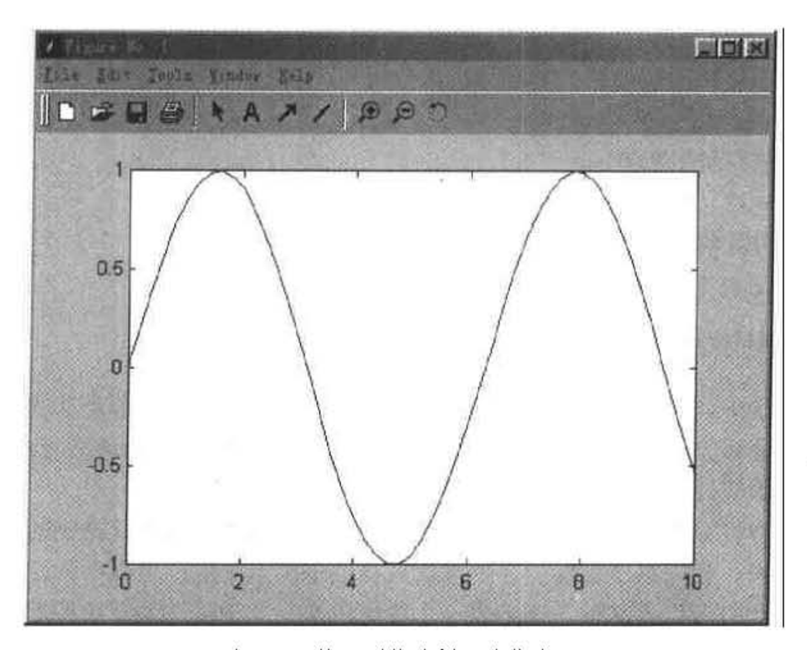


图 3-44　按返回值绘制正弦曲线

3.6.3　使用 out1 模块将仿真输出信息返回到 MATLAB 命令窗口

【例 3-9】（续【例 3-8】），使用输出“out1”模块返回数据信息到 MATLAB 命令窗口中。

在信号与系统“Signals & System”模块库中，有一个名为“Out1”的输出模块。可以将系统仿真结果的信息输入到这个模块，如图 3-45 所示。该输出模块会将数据返回到 MATLAB 命令窗口中，并自动用一个名为“yout”的变量保存起来。MATLAB 也会自动将每个时间数据存入 MATLAB 命令窗口中，用“tout”这个变量保存起来。

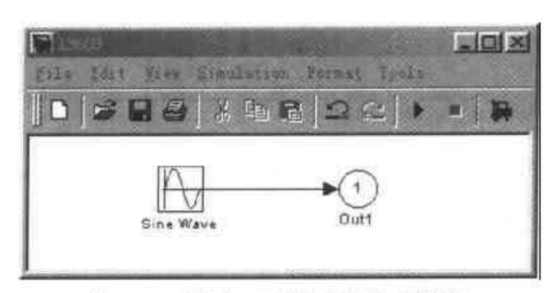


图 3-45　使用 out 模块返回数据信息

将控制系统输出数据与时间数据都返回到 MATLAB 命令窗口之后，也可以用绘图命令在 MATLAB 命令窗口里绘制出图形。指令如下：

```
plot (tout,yout)
```

这个指令执行后，可以看到，所绘制的图形也与图 3-44 完全一样。

3.7　SIMULINK 的分析工具

SIMULINK 中一般使用的仿真分析工具有：

① 模型线性化 。

②求控制系统平衡点。

③绘制波德（Bode）图。

④绘制时间响应曲线。

⑤ 转换 LTI 对象等。

3.7.1　模型线性化概述

实际工程中，理想的线性系统是不存在的，绝大多数元器件都是非线性的。能否用线性系统的分析方法来研究非线性系统呢？答案是肯定的。研究的思路与办法多种多样，非线性系统的线性化就是办法之一。

所谓非线性系统的线性化就是对一个非线性系统的模型找出其稳定的平衡点，如果在工作过程中，代表系统属性的各物理量只在该平衡点附近产生微小的变化，非线性模型就能够以此平衡点为基础，表示成一个线性模型，关于线性系统的控制理论都能适用于该模型。这便是自动控制理论里关于小偏差线性化方法或称增量线性化方法的概念。

但是将非线性系统表示成一个线性系统，首先涉及到控制系统在工作过程中各状态物理量是否只在该平衡点附近产生微小变化；其次还会牵涉到很多复杂的数学理论，比如非线性特性在平衡点附近是否连续、可导以及求解偏微分方程等问题。而 MATLAB 恰好提供了特别的函数命令专门解算模型的线性化问题。模型线性化包括连续系统与离散系统两类线性化模型。

3.7.2　连续系统的线性化模型

对于非线性系统有状态方程：

$$\begin{cases}\dot{x}=f\left(x\left(t\right),u,t\right)\\ y=h\left(x\left(t\right),u,t\right)\end{cases}\qquad(3\text{-}35)$$

如果在某平衡工作点 $\boldsymbol{x}$、输入 $\boldsymbol{u}$ 与时间 t 指定的条件下，将其表示成状态空间模型：

$$\begin{cases}\dot{x}=Ax+Bu\\ y=Cx+Du\end{cases}\qquad(3\text{-}36)$$

就可以用 SIMULINK 提供的 linmod 或 linmod2 函数命令将式（3-35）描述的非线性系统在某平衡工作点表示为近似的线性模型。状态空间线性模型关系式（3-36）中：

$\boldsymbol{x}$ ——系统 n 维状态向量；

$\boldsymbol{u}$ ——系统 r 维输入向量；

$\boldsymbol{y}$ ——系统 m 维输出向量；

$\boldsymbol{A}$ ——$n\times n$ 的状态矩阵，由控制对象的参数决定；

$\boldsymbol{B}$ ——$n\times r$ 的控制矩阵；

$\boldsymbol{C}$ ——$m\times n$ 的输出矩阵；

$\boldsymbol{D}$ ——$m\times r$ 的直接传输矩阵。

linmod 或 linmod2 函数指令调用的一般格式为：

[***A***, ***B***, ***C***, ***D***]= linmod 或 linmod2 （‘模型名称’，工作点向量 ***x***, 输入向量 ***u***）

SIMULINK 提供的“linmod” 和“ linmod2”两个函数，可以从连续系统中提取线性模型，也就是两个函数命令执行后，都可以得到一个用[***A***,***B***,***C***,***D***]表达的状态空间模型。它们的调用格式如下：

[***A***, ***B***, ***C***, ***D***] = **linmod**（*'modelname'*, ***x***, ***u***, ***para***, ***xpert***, ***upert***）

[***A***, ***B***, ***C***, ***D***] = **linmods2** （*'modelname'*, ***x***, ***u***, ***para***, ***apert***, ***bpert***, ***cpert***, ***dpert***）

其中：

（1）第一个输入参数*'modelname'*为模型名称，是必不可少的，它是被仿真系统的动态结构方框图模型的文件名。动态结构图模型文件名可以省略扩展名，不影响函数的使用也不影响调用函数之程序的使用。

（2）第二、第三个输入参数 ***x*** 和 ***u*** 分别为系统的工作点状态和输入信号。缺省设置值为全零向量。

（3） 第四个输入参数 ***para*** 为二元向量。***para*** （2） 指定时间点，缺省设置值为 0。***para***（1）指定计算时所用的扰动值，在“linmod” 中，缺省设置值为 1e-5；在“ linmod2”中，缺省设置值是 1e-8。允许缺省。

（4） 输入参数 ***xpert*** 和 ***upert*** 分别设置状态分量和输入分量的扰动值，允许缺省。

（5） ***apert***, ***bpert***, ***cpert***, ***dpert*** 用来设置系统状态与输入的综合扰动值。

需要指出，虽然 linmod2 （ ） 函数命令执行后的结果比 linmod （ ） 的准确，但是它运行的时间较长。

3.7.3 离散系统的线性化模型

SIMULINK 提供的 dlinmod（ ） 函数能够从离散、多频或混合系统中提取一个任何给定采样频率的近似线性模型。当采样时间 *ts* 取为零时，就可得到近似的连续线性模型；

否则，得到离散线性模型。该指令的一般调用格式是

[*Ad*, *Bd*, *Cd*, *Dd*] = **dlinmod** （'*modelname*', *ts*, *x*, *u*, *para*）

其中：

（1）在任何调用方式中，第一、第二输入参数是必不可少的，*ts* 是指定的采样时间。

（2）在原系统稳定的前提下，若 *ts* 是原系统所有采样周期的整数倍，或者 *ts* 不小于原系统中最慢的采样周期，则由“dlimod”所得线性模型在 *ts* 采样点上与原系统有相同的频率响应和时间响应。

（3）输入参数 ***x***、***u*** 和 ***para*** 的解释同连续系统的线性化模型。

3.7.4 SIMULINK 平衡点的求取

工程上往往以非线性系统在某平衡点的数据信息为依据来分析系统的稳定性。所谓平衡点是指所有状态向量的导数都为 0 的点。仅有部分状态向量的导数为 0 的点则叫做偏平衡点。在系统的（偏）平衡点，系统的状态（或部分状态）变化最小。SIMULINK 中对于非线性系统求取平衡点的功能是通过 trim（ ）函数命令实现的。

trim（ ）函数命令求取平衡点的基本原理是利用数值计算，寻找这样一个点，该点使得状态向量的导数为最小。trim（ ）函数命令的运行有两种可能结果：一种是寻找的平衡点就是或者最接近初始值，另一种是寻查不到平衡点，寻找失败。

trim（ ）函数命令的调用格式为：

[*x*, ***u***, *y*, *dx*, *option*] = **trim**（'*modelname*', *x0*, *u0*, *y0*, *ix*, *iu*, *iy*,*dx0*,*idx*,*options*,*t*）

其中：

（1）*modelname* 是系统模型文件名，这个输入参数是必不可少的。

（2）*x0*, *u0*, *y0* 是开始寻找点的状态、输入、输出的初始猜测值；*ix*, *iu*, *iy* 是用来分别指定 *x0*, *u0*, *y0* 中保持不变的分量下标。

（3）*idx* 用来指定导数不为 0 的状态分量，而 *dx0* 用来指定导数不为 0 的导数值。

（4）*options* 是优化计算方法的参数设置。*t* 用来指定计算时间。

需要指出，若想寻找最佳平衡点，不妨多尝试或运算几组初始预测值，加以比较后取其效果最佳者。

3.7.5 SIMULINK 里由状态方程转换成 LTI（线性时不变系统）对象

在 SIMULINK 里，可以将用矩阵组[***A***,***B***,***C***,***D***]描述的状态空间模型转换成 LTI（线性时不变系统）对象。其命令格式如下：

sys = **ss**（***A***,***B***,***C***,***D***）

3.7.6 在 SIMULINK 里绘制波德（Bode）图与时间响应图

波德（Bode）图就是控制系统的对数振幅频率特性曲线与对数相位频率特性曲线，它是自动控制中频域分析的常用工具（详见第四章）。SIMULINK 里提供绘制波德图的函数命令格式为：

bode（*sys*）或 **bode**（***A***,***B***,***C***,***D***）

命令格式中的（***A***，***B***，**C**，***D***）就是描述状态空间模型的矩阵，*sys* 则是系统的开环传递

函数。命令执行后即可自动绘制出系统的波德图。SIMULINK 里还提供自动计算频域性能指标的函数命令。其命令格式为：

margin（*sys*）

命令执行后即可求出系统的频域性能指标，还可以将频域性能指标附在波德图上。

时域响应图是自动控制中时域分析的重要工具（详见第 4 章）。SIMULINK 里提供的求系统阶跃响应曲线的函数命令格式为：

step（*sys*）或 **step**（*A,B,C,D*）

必须指出，命令格式中的 ***sys*** 是系统的闭环传递函数，而 *A*，*B*，*C*，*D* 则是描述闭环系统的状态空间模型。命令执行后即可自动绘制出系统单位阶跃响应曲线。

在 MATLAB 5.3 版本里，这些曲线绘制时，其纵、横坐标的刻度与范围是智能化的，系统会自动设定好，一般不须人为设置。

【例 3-10】双环调速的电流环系统的方框图模型如图 3-46 所示。试求其线性模型。

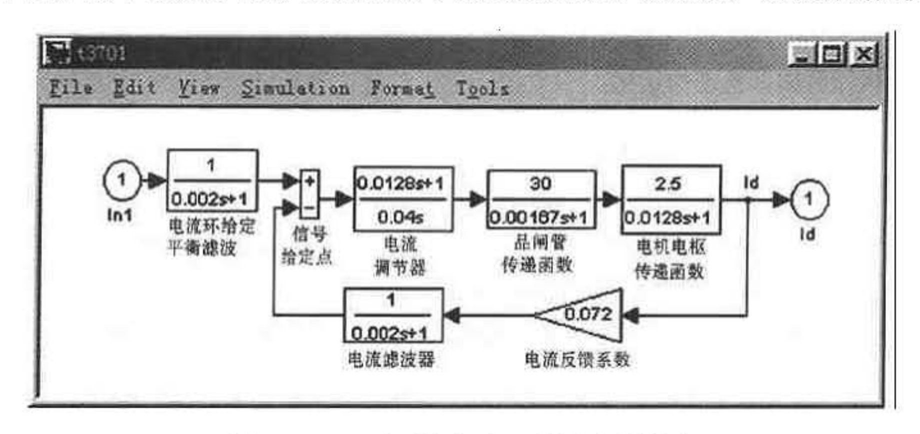


图 3-46 双环调速的电流环系统方框图模型

【解】

（1）建立模型

在 SIMULINK 中建立如图 3-46 所示的模型。

（2）求状态空间模型

运行以下指令，可得到线性系统的一个线性状态空间模型的[A,B,C,D]描述。

```
cd c:\matlabr11\work
[A,B,C,D]=linmod('t3701')
A =
  1.0e+004 *
   -0.0078         0         0         0    1.7964
         0   -0.0500         0         0         0
    0.0014         0   -0.0500         0         0
         0    0.0500   -0.0500         0         0
         0    0.0160   -0.0160    0.0025   -0.0599
B =
```

```
      0
      1
      0
      0
      0
C =
  195.3125  0  0  0  0
D =
      0
```

（3）求传递函数

运行以下命令，即可得到闭环系统的传递函数并加以显示：

```
[num,den]=ss2tf (A,B,C,D);
printsys (num,den,'s')
```

命令执行后加以整理的结果：

```
num/den =
    561377245.509s^2+324546220059.88s+21928798652694.6
----------------------------------------------------------------
s^5 + 1676.9274 s^4 + 9737+e02s^3 + 2564e+05s^2 + 3506e+07 s+ 1579e+09
```

（4）求系统的频率特性曲线及阶跃响应曲线

当运行以下程序后，即可得到系统的 Bode 图（图 3-47）及阶跃响应曲线（图 3-48）。

```
% MATLAB PROGRAM L3701.m
%
sys1=tf([0.0128 1],[0.04 0]);
sys2=tf([0 30],[0.00167 1]);
sys3=tf([0 2.5],[0.0128 1]);
sys4=tf([0 0.072],[0.002 1]);
sys0=tf([0 1],[0.002 1]);
figure(1);
margin(sys1*sys2*sys3*sys4);hold on
figure(2)
closys=sys0*(sys1*sys2*sys3/(1+sys1*sys2*sys3*sys4))
step(closys)
```

关于本题的求解还须说明：

（1）绘制控制系统的 Bode 图并求系统的频域性能指标时，必须使用系统的开环传递函数。这从以上的 MATLAB 程序中已经可以看出。

（2）求解控制系统的阶跃时间响应时，是针对在单位阶跃信号作用下,计算整个系统即闭环系统的输出量随时间变化的曲线，必须使用系统的闭环传递函数。这也可以从以上的 MATLAB 程序中看出。

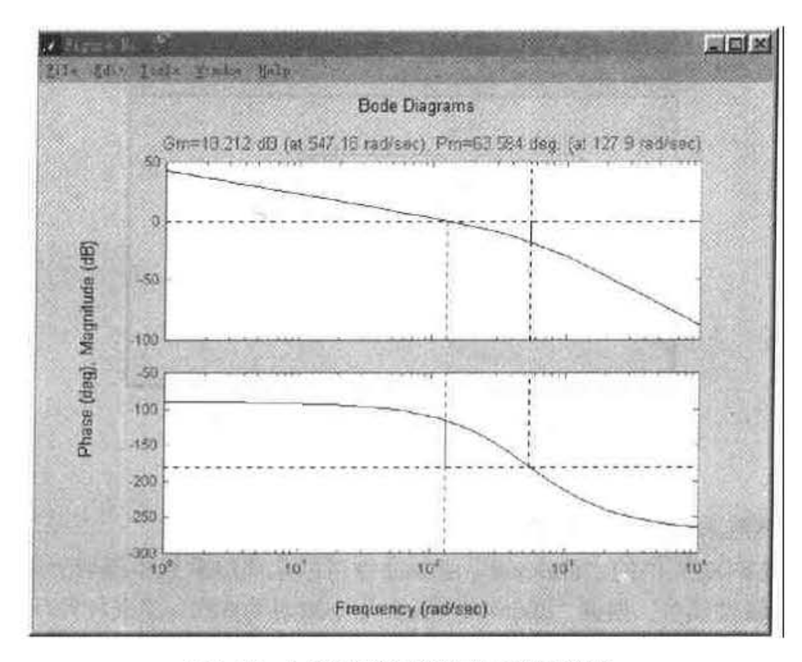


图 3-47　电流环系统的波德图与频域指标

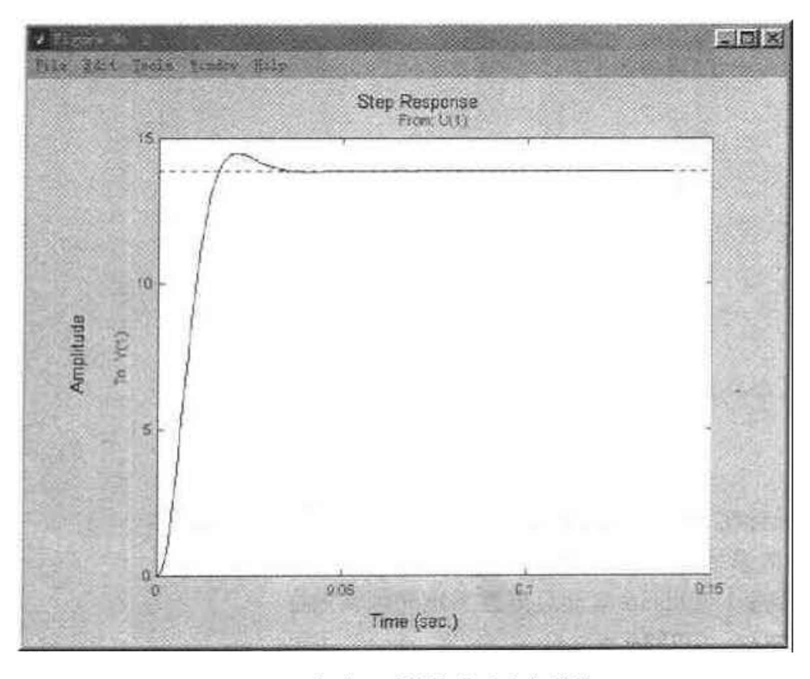


图 3-48　电流环系统的阶跃响应曲线

（3）在算得的传递函数中，分子 s^4 项与 s^3 项的系数为 1e-013 以上数量级（传递函数表达式中未计入），这是由计算截断误差产生的，它实际为零。故以上所求传递函数式 $\frac{num}{den}$ 中，分子未给出 s^4 与 s^3 项。

【例 3-11】求图 3-49 所示的离散系统的线性模型、频域性能指标与时域响应。

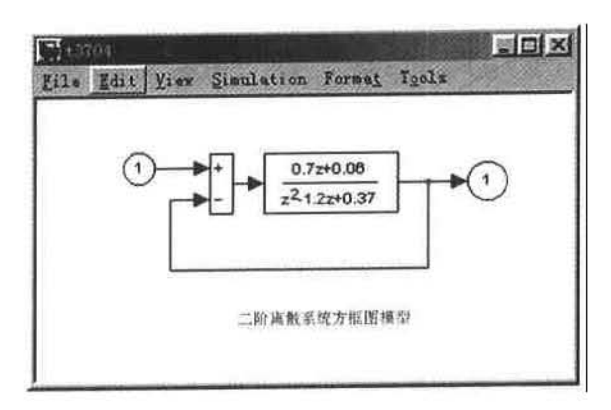


图 3-49 二阶离散线性系统方框图模型

【解】

（1）求离散系统的线性模型

由 SIMULINK 提供的“dlinmod”函数命令可以从离散系统中提取一个任何给定采样频率的近似线性模型。根据“dlinmod”函数命令的调用格式，当采样时间为 ts=0.1 时，用以下语句提取一个近似线性模型。

```
cd c:\matlabr11\work
[A,B,C,D]=dlinmod('t3704',0.1)
```

语句执行后可得状态空间线性模型：

```
A =
    0.9985    -0.0800
    0.1861     0.9055
B =
    0.0876
   -0.0985
C =
    0.7000     0.0600
D =
0
```

当改变采样时间时，求出的状态空间线性模型数据也会变化。

（2）离散系统的频域计算

用以下程序 L3702a.m 来计算离散系统的频域指标。

```
% MATLAB PROGRAM  L3702a.m
%
num=[0.7 0.06];
den=[1 -1.2 0.37];
[mag,phase,w]=dbode(num,den,0.1);
[Gm,Pm,Wcg,Wcp]=margin(mag,phase,w)
```

程序运行后可得系统的频域指标：

$Gm = 4.0156$ 即系统模值稳定裕度 $L_h = 20\log10(4.0156) = 12.075\ \text{dB}$

$Pm = 43.5423$ 即系统相位稳定裕度 $\gamma = 43.5423°$

$Wcg = 31.4159$ 即系统-π 穿越频率 $\omega_{cg} = 31.4159\ \text{rad/sec}$

$Wcp = 10.2979$ 即系统剪切频率 $\omega_{cp} = 10.2979\ \text{rad/sec}$

（3）求离散系统的时域响应

用以下程序 L3702b.m 来计算离散系统的时域响应曲线。

```
%MATLAB PROGRAM  L3702b.m
%
num1=[0.7 0.06];
den1=[1 -1.2 0.37];
num2=1;
den2=1;
[num,den]=feedback(num1,den1,num2,den2);
dstep(num,den)
```

程序运行后可得如图 3-50 所示的系统的单位阶跃响应曲线。

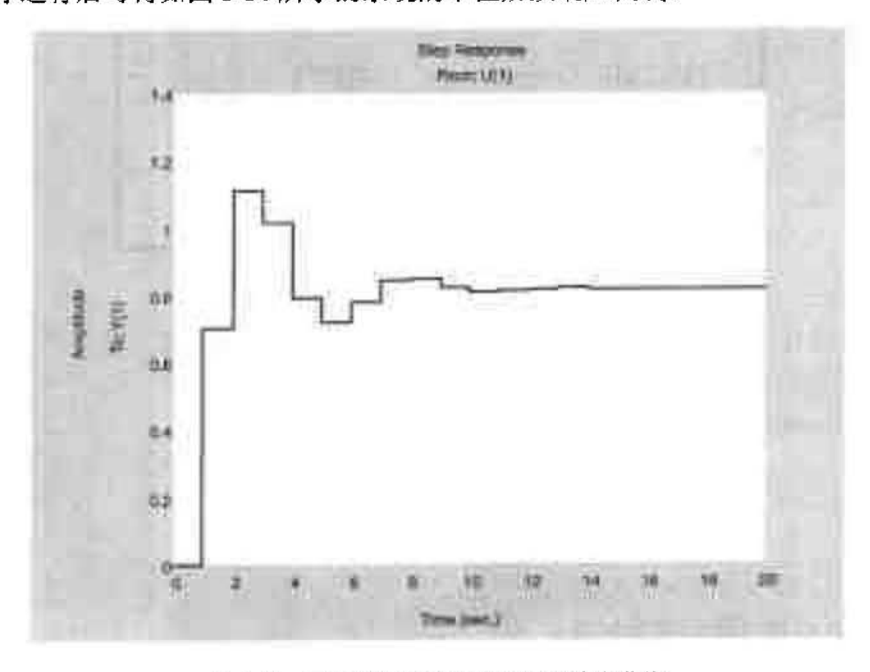


图 3-50　二阶离散系统的单位阶跃响应曲线

【例 3-12】求图 3-51 所示的非线性系统的线性模型、平衡工作点、线性状态空间模型对应的传递函数，将其转换成 LTI 对象，并求系统的 Bode 图及阶跃响应曲线。图 3-51 中非线性饱和模块（Saturation）使用的上下限设定为+1 与-1（缺省设置为 +0.5 与 -0.5）。

【解】

（1）运行以下指令，可得到非线性系统的一个线性状态空间模型的[A,B,C,D]描述。

```
cd c:\matlabr11\work
```

```
[A,B,C,D]=linmod ('t3706')
A =
   -3.0000   -2.0000    2.8000
    1.0000         0         0
    0        -0.5800         0
B =
     0
     0
     1
C =
     0    0.5800   0
D =
     0
```

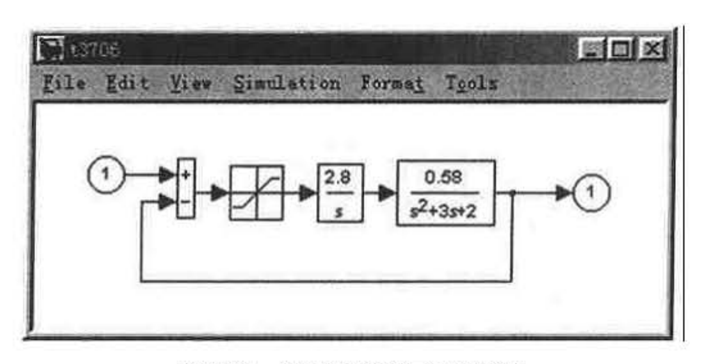


图 3-51　非线性系统方框图模型

（2） 运行以下指令，可得到线性状态空间模型对应的传递函数：

```
[num,den]= ss2tf (A,B,C,D);
printsys (num,den,'s')
num/den =
   -1.7764e-015 s^2 - 3.5527e-015 s + 1.624
   ----------------------------------------
    s^3 + 3 s^2 + 2 s + 1.624
```

（3）运行以下语句求平衡点：

```
cd c:\matlabr11\work
x = [0;0;0];u = 0;y = [1];
ix = [ ];
iu = [ ];
iy = [1];
[x,u,y,dx] = trim ('t3706',x,u,y,ix,iu,iy)
```

指令执行后，可得平衡点的信息：

```
x =
```

```
    0.0000
    1.7241
    1.2315
u =
     1
y =
    1.0000
dx =
  1.0e-015 *
         0
    0.0000
   -0.2220
```

（4）使用如下命令将用[A,B,C,D]描述的状态空间模型转换成 LTI（线性时不变）对象。

```
sys = ss  (A,B,C,D)
a =
          x1     x2     x3
    x1    -3     -2    2.8
    x2     1      0      0
    x3     0  -0.58      0
b =
          u1
    x1     0
    x2     0
    x3     1
c =
          x1     x2     x3
    y1     0   0.58      0
d =
          u1
    y1     0
Continuous-time model.
```

运行结果表明，LTI（线性时不变）模型就是状态空间模型。

月以下程序 L3703.m 来计算统的 Bode 图及阶跃给定响应。

```
% MATLAB PROGRAM L3703.m
%
sys=tf(2.8,[1 0]);
sys1=tf([0.58],[1 3 2]);
figure(1);
```

```
margin(sys*sys1);hold on
figure(2)
closys1=sys*sys1/(1+sys*sys1);
step(closys1)
```

运行程序后可得系统的 Bode 图（如图 3-52 所示）及单位阶跃给定响应曲线（如图 3-53 所示）。

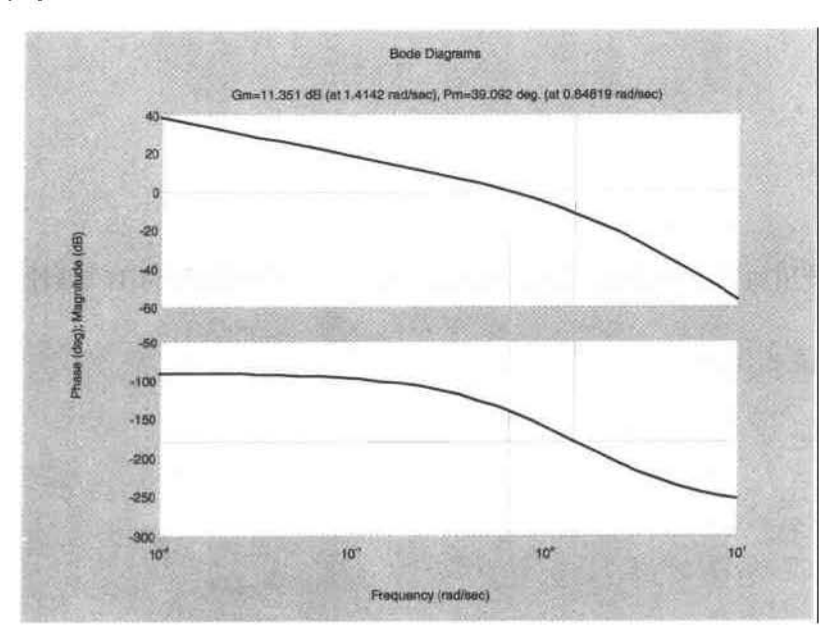


图 3-52　非线性系统波德图

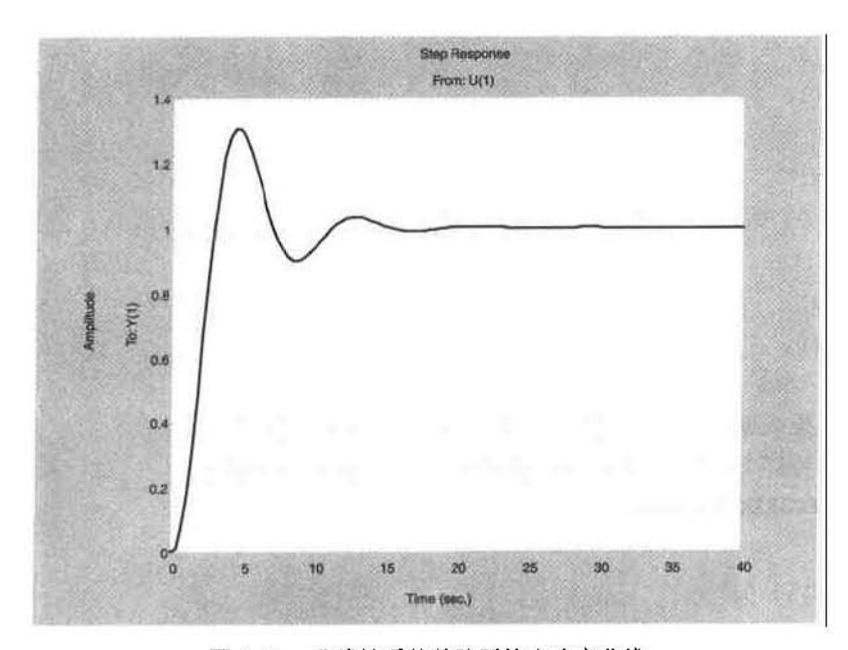


图 3-53　非线性系统的阶跃给定响应曲线

关于本题的求解还须说明：

（1）最后一项输出 dx 的数量级为 1e-015，也是截断误差产生的，实际应为零。在不同的机器上，得出的数值会不一样，但数量级不变。

（2）在绘制控制系统的 Bode 图以及求系统的频域性能指标时，必须使用系统的开环传递函数。

（3）求解闭环控制系统的阶跃时间响应时，必须使用系统的闭环传递函数。

3.8 SIMULINK 3.0 的自定义模块库与自定义子系统

SIMULINK 3.0 提供了自定义模块库与自定义子系统的功能，这两项功能都有实际意义与实用价值。这一节就介绍自定义模块库与自定义子系统的概念、方法与步骤。

3.8.1 自定义模块库与自定义子系统概述

当做了许多系统模型的建立与分析以后，会发现每次创建一个新的系统模型时，许多常用的模型要不断地重新建立，这样非常不方便；或者觉得 SIMULINK 原有的 8 类基本模块库太多，常常有许多门类模块或者某一类里的很多标准功能模块，对于解决一个特定的问题时根本用不到。基于这些原因，SIMULINK 提供了自定义模块库的功能。根据解决问题的不同需要，自定义模块库既可以由选择 8 类基本模块库中的几类构成，也可以由 8 类基本模块库里再选择其中多个标准功能模块构成。

控制系统建立的模型型式各种各样，功能千差万别。但是经常需要将模块库里的多个模块组合（Group）起来，构成一个新的功能模块。这就是自定义新功能模块。

自定义新功能模块有两种方法：一种方法是采用 SIMULINK 3.0 库浏览器的“Signals & systems” 模块库中的 “Subsystem” 功能模块，利用其编辑区重新设计组合新的功能模块。这种方法是先拷贝“Subsystem”子系统的外套（罩），然后打开“Subsystem”子系统窗口，将要组合的功能模块拷贝其中；另外一种方法是在模型窗口中执行菜单【Edit】下的【Greate Sunsystem】命令，创建新功能模块。这种方法是先有由多个标准基本模块组合成的子系统内容，然后再将其创建子系统，即套以“Subsystem”的外套（罩）。

这两种方法都非常实用且有效。对于复杂、大型系统的模型，这些措施可以减少功能模块的个数，简化图形，使控制系统仿真模型的层次结构更加清晰，更方便大型复杂系统的仿真。

对于复杂的大型控制系统，创建模块化的结构图模型是最佳的形式，因为这样的模型结构清晰，上下层次分明，相互关系明确。为实现模块化的要求，SIMULINK 3.0 提供了定义子系统的功能，可把能够实现某些功能的相关模块组合在一起，成为子系统。

3.8.2 SIMULINK 窗口下自定义模块库

在 SIMULINK 3.0 模型窗口里主菜单【File】下执行【New/Library】菜单命令，会出现一个新的 Library 窗口，名为“Library：untitled”，如图 3-54 所示。



图 3-54　Library：untitled 窗口

当启动 SIMULINK 时，无论是打开模块库浏览器图 3-3 还是打开八类基本模块库图 3-4，这些模块库都是“只读性”的，无法修改其中的内容，也就是被锁住的，那怕只是将模块图标移动一下都不可能。因此要先将其改锁。这可以选择【Edit】菜单下的【Unlock Library】命令，执行后就可以修改其中的内容了。

为了说明自定义函数模块库的过程，请看下例。

【例 3-13】建立一个只有“Continuous”、“Math”、“Sources”与“Sinks”四类模块库组成的自定义模块库 Library1；同时再建立一个只有“Gain”、“Derivative”、“Integrator”、“In1”、“Out1”、“Transfer fcn ”、“Sum”与“Zero-pole”等 8 个基本模块构成的自定义模块库 Library2。

【解】

1．建立自定义模块库 Library1

（1）在 SIMULINK 3.0 模型窗口里主菜单【File】下执行【New/Library】菜单命令，会出现一个新的 Library 窗口，命名为“Library：Library1”，如图 3-55 所示。

（2）将 SIMULINK　3.0　提供的 8 类基本模块库中的“Continuous”、“Math”、“Sources”与“Sinks”四类模块库图标，从模块库浏览器中分别拖拽到图 3-55 所示“Library：Library1”里，并存盘，请注意存盘的路径。

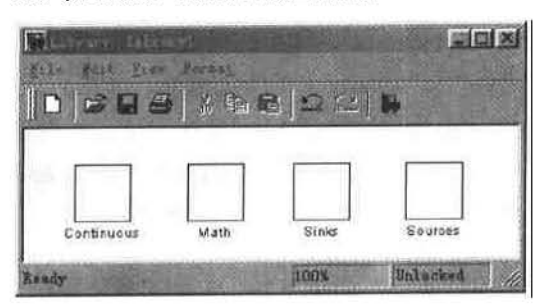


图 3-55　自定义函数模块库 Library1

这样自定义的模块库 Library1 是“只读性”的，若执行修改其中内容的操作，系统会给出如下图 3-56 的提示。



图 3-56　未解锁时的“只读性”提示

如果执行【Edit】菜单下的【Unlock Library】菜单命令，就可以修改其中的内容了。

2．建立自定义模块库 Library2

（1）在 SIMULINK 3.0 模型窗口里主菜单【File】下再次执行【New/Library】菜单命令，会出现一个新的 Library 窗口，命名为“Library：Library2”，如图 3-57 所示。

（2）将连续系统模块（Continuous）中的“Integrator”、“Derivative”、“Transfer fcn ”与“Zero-pole”四个标准基本模块图标，分别拖拽到图 3-57 所示的“Library：Library2”里，并存盘，请注意存盘的路径。

（3）重复以上操作，将数学运算模块（Math）中的“Gain”与“Sum”两个标准基本模块图标，再次分别拖拽到“Library：Library2”里，并存盘。

（4）再次重复以上操作，将信号与系统（Signals & System）中的“In1”与“Out1”两个标准基本模块图标，分别拖拽到图 3-57 里，并存盘。

最后自定义模块库 Library2 的结果如图 3-57 所示。同样自定义的模块库 Library2 是“只读性”的，执行主菜单项【Edit】下【Unlock Library】的菜单项命令，即可修改其中的内容。

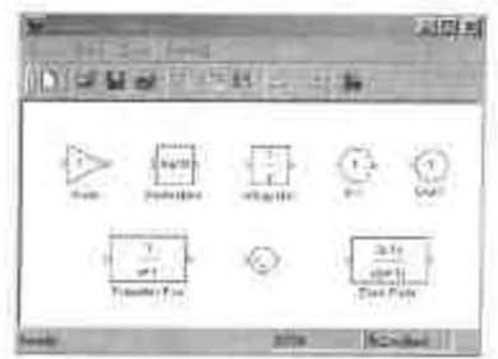


图 3-57　自定义函数模块库 Library2

3.8.3　SIMULINK 窗口下自定义子系统

采用两种方法，其一是采用 SIMULINK 3.0 库浏览器“Signals & systems” 模块库中的“Subsystem” 标准功能模块，其二是在模型窗口中执行菜单【Edit】下的【Greate Sunsystem】命令的方法，无论那一种方法都可以自定义子系统。

3.8.3.1　采用 SIMULINK 的“Subsystem”功能模块自定义子系统

为了说明用标准功能模块“Subsystem”来自定义子系统的过程，请看以下举例。

【例 3-14】用 SIMULINK 的“Subsystem”功能模块将直流调速系统的电流环自定义为子系统，并对其进行单位阶跃给定响应的仿真。

1．定义子系统

（1）先到 Simulink 库浏览器的“Singnals & System”模块库中把标准功能模块“Subsystem”复制到空白模型设计区域（“untitled”）内，还复制功能模块“Step”与“Scope”模块。再完成以下两步后，模块“Subsystem”即会生成输入与输出接口，将其连成系统如图 3-58 所示，模型命名为“t3805.mdl”后存盘。

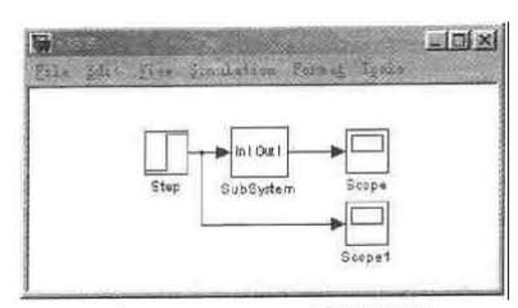


图 3-58 Subsystem 子系统模型

（2）鼠标左键双击模块“Subsystem”，就将会弹出“t3805/Subsystem”的设计区。

（3）在“t3805/Subsystem”的设计区域（图 3-59）中进行直流调速系统电流环的设计（拷贝各个标准功能模块、连线构成系统模型）或者将设计好的电流环模型拷贝其中，并在输入与输出端加入“In1”和“Out1”两个功能模块（图 3-59）。模型设计完成后，请注意存盘。至此子系统创建完成。

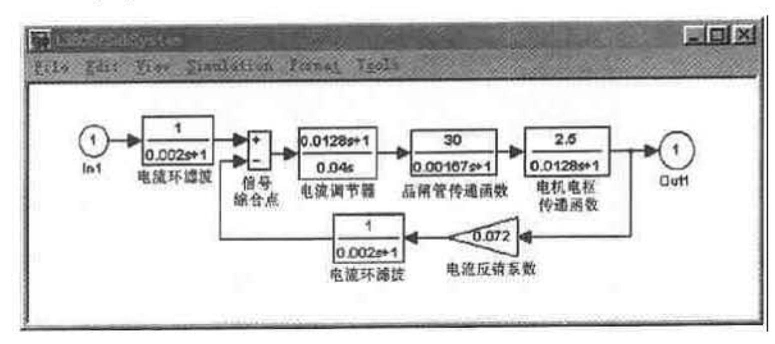


图 3-59 电流环的仿真模型

2．进行电流环单位阶跃给定响应仿真

（1）选择主菜单的【Simulation】下的【Start】菜单命令，执行后听到“嘟”声，仿真过程即结束。

（2）用鼠标左键双击“Scope”与“Scope1”，即可看到单位阶跃给定响应仿真曲线如图 3-60 与单位阶跃给定信号如图 3-61。为了能够看到阶跃信号的前沿，特意将两图信号的起始作用时间设定在 1 秒处。

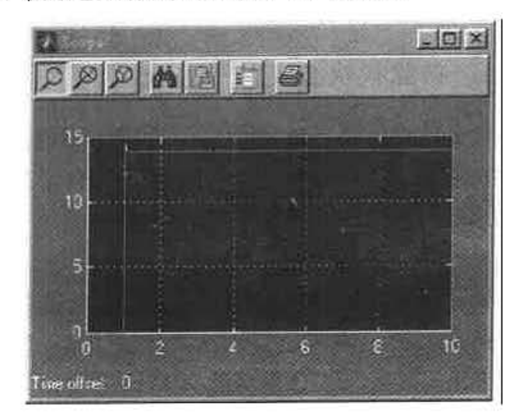


图 3-60 电流环阶跃给定响应仿真曲线

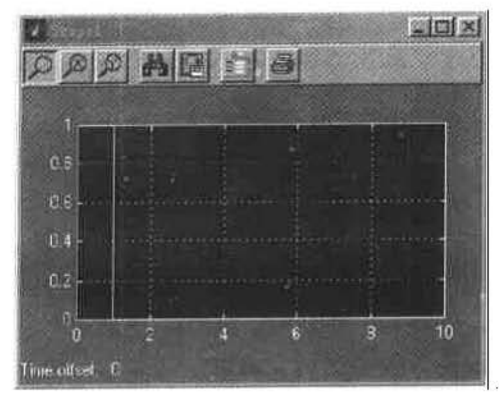


图 3-61 单位阶跃给定信号

3.8.3.2 采用 SIMULINK 的 Creat Subsystem 命令自定义子系统

为说明在模型窗口中执行菜单【Edit】下的【Greate Sunsystem】命令自定义子系统的方法，特举下例说明。

【例 3-15】用 SIMULINK 模型窗口菜单【Edit】下的【Creat Subsystem】命令，对于双闭环调速系统（图 3-62）的电流环定义其子系统，并对双环系统进行单位阶跃给定响应的仿真。

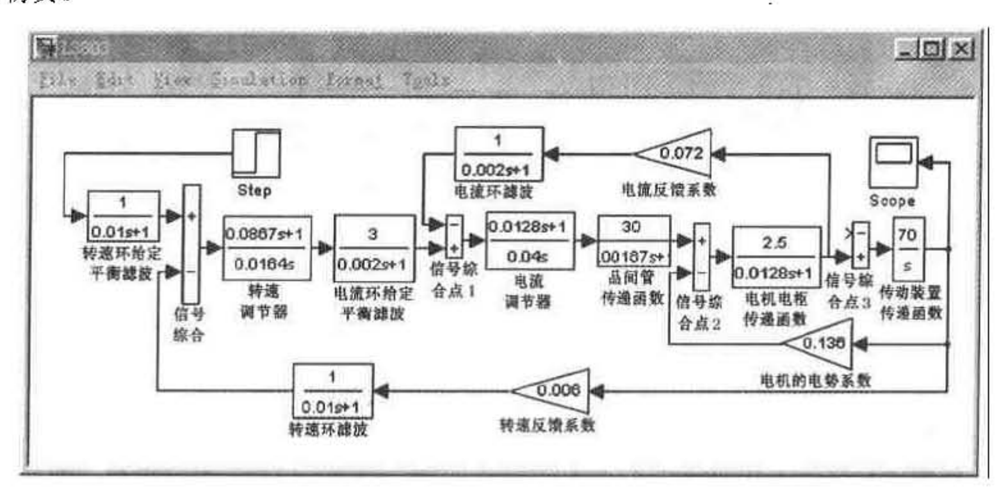


图 3-62　双环调速系统

【解】

图 3-62 所示是直流拖动双闭环调速系统。为简化其模型方框图，使模型结构层次更加清晰，欲将系统中的电流环部分定义为一个子系统，其操作步骤如下：

（1）在图 3-62 里，按住鼠标左键并拖拽形成虚线框将要定义子系统的电流环部分框住。被框住的模块与其连线将被选中。

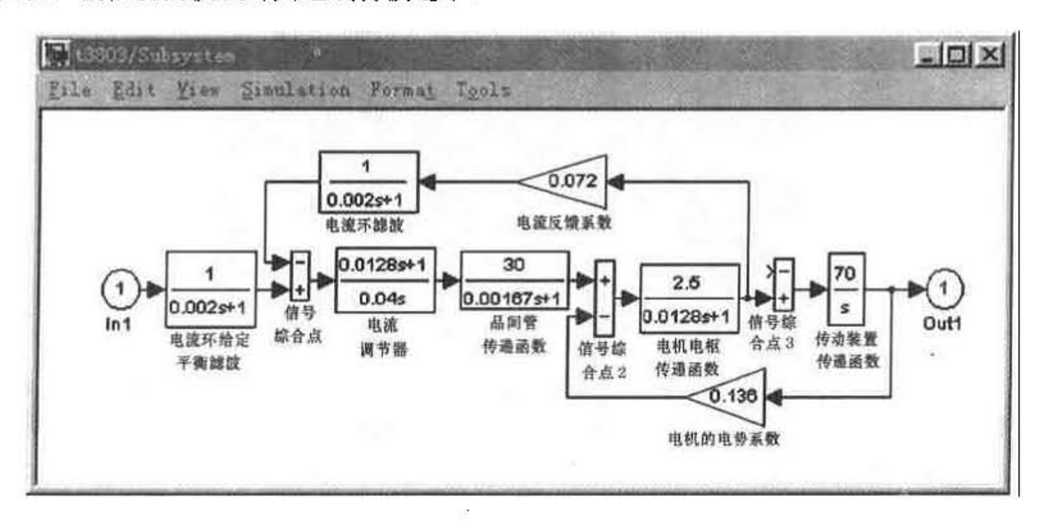


图 3-63　被定义的带交叉反馈电流环的"Subsystem"子系统

（2）在图 3-62 的模型窗口里，选择【Edit】菜单下的【Create Subsystem】命令，执行后会弹出图 3-63 所示被定义的子系统模块“L3803/Subsystem”，请注意，图 3-63 中必须再添加“In1”与“Out1”功能模块。

（3）再打开原双环系统的模型窗口，会发现被虚线框住的电流环部分已被“Subsystem”的新图标所代替，参见图 3-64。至此，子系统创建成功。

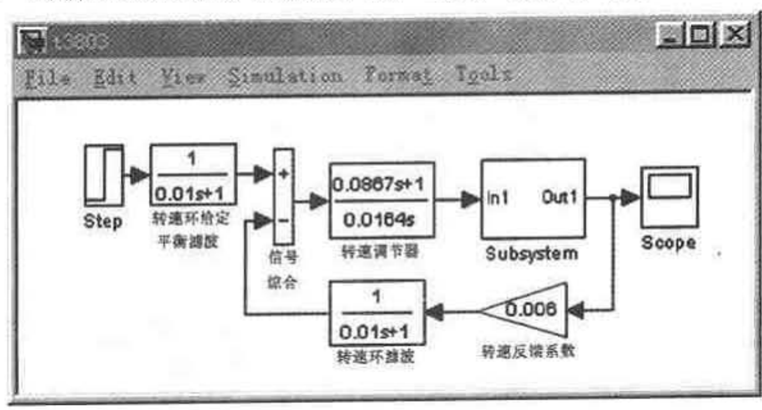


图 3-64 带“Subsystem”子系统的双环调速系统

（4）对于图 3-64 所示系统进行单位阶跃给定响应仿真，可以就在其模型窗口里操作。选择主菜单项【Simulation】下【Starte】的菜单项命令，执行后可听到“噹”声，再用鼠标左键点击示波器“Scope”图标，即可得图 3-65 所示系统单位阶跃给定响应曲线。

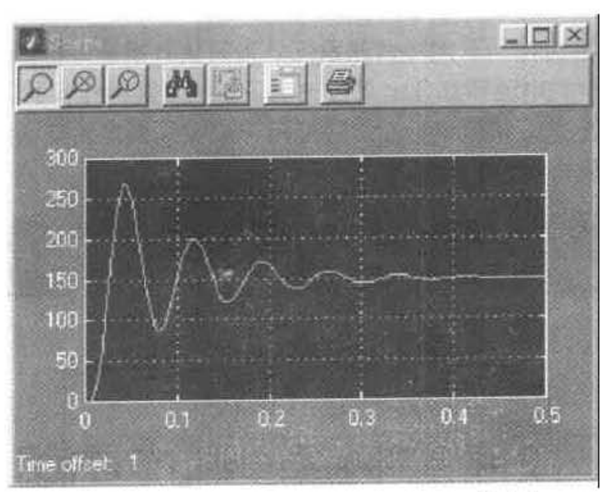


图 3-65 系统单位阶跃给定响应衰减振荡曲线

必须指出，为了能观察到曲线的最佳显示效果，针对具体系统，须要恰当设置仿真参数与示波器参数，否则，在示波器里看不到满意的曲线。有关仿真参数与示波器参数的设置已经在 3.5、3.6 中介绍过，此处不再赘 述。

如果改变系统中转速环前向通道的增益，例如减小为原来的确 1/3，系统的单位阶跃给定响应将无振荡，如图 3-66 所示。

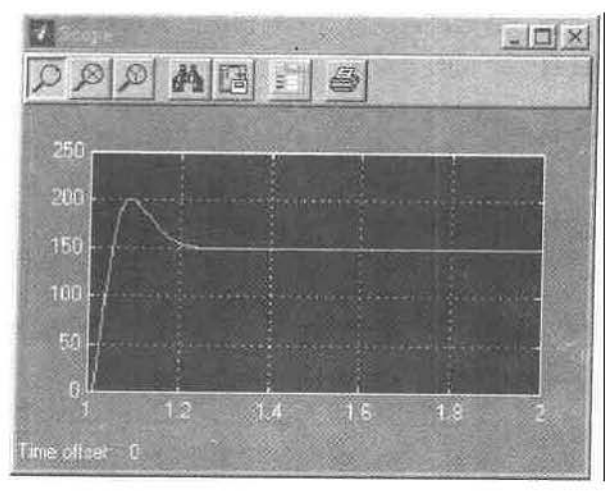


图 3-66　系统无振荡单位阶跃给定响应曲线

3.9　SIMULINK 的 Mask 功能

将所需要的多个标准基本模块，采用 SIMULINK 的封装功能，将其集成在一起，形成新的功能模块——特殊的子系统，这就是 SIMULINK 的 Mask 功能。Mask 功能是一种对子系统（Subsystem）的处理措施。经封装后的子系统，可以有特定的图标与参数设置对话框，成为一个独立的功能模块。事实上，在 SIMULINK 的模块库里有许多标准模块（如 PID ）本身就是由多个更基本的标准功能模块封装后构成的。

3.9.1　Mask（封装）概述

已经介绍过，当创建一个复杂的系统模型后，为了能够制作成一个结构层次分明的系统，并将功能相关的模块组合在一起，以实现模块化管理的要求，常常会使用“Subsystem”子系统功能模块，建立许多子系统。但是往往子系统一多，管理起来会变得相当麻烦。原因是当需要修改子系统特别是要改变子系统内多个模块的参数时，就要逐个打开模块参数对话框来进行操作。如果要修改的模块参数又很多时，修改的工作就会变得相当繁琐。

为了解决这个问题，SIMULINK 提供了一个“Mask”— 称之为“封装”的功能，可以将功能相关的基本模块集成并封装在一起成为一个特殊的“Subsystem”，并为之定制一个对话框，将子系统内众多的模块参数对话框集成为一个完整的对话框，就像定制一个特殊的模块一样。

任何一个“Subsystem”子系统模块都可以进行“Mask”（封装）。

封装子系统模块并定制其对话框能够方便用户使用以及提高仿真效率。具体优点如下：首先，将子系统的多个模块参数对话框集成为一个对话框，可在该对话框内输入或者修改同属于一个子系统的不同模块参数。使用封装定制的参数对话框，可以避免不小心修改了不该改变的参数。其次，可以制作被封装子系统的图标，用来形象标识被封装

了系统模块并说明其用途。最后，可以将对个别模块的描述或者帮助集成在一起，这样能够有效地帮助了解被封装定制的子系统。

3.9.2 Mask 编辑器

Mask 即“封装”是在“Mask Editor”（Mask 编辑器）里进行的。图 3-67 就是打开 Icon 标签页的 Mask 编辑器。所谓“封装”，就是将“Mask Editor”里三个标签页的相关信息栏按照要求填入规定的信息及数据，这也叫做定制参数对话框。

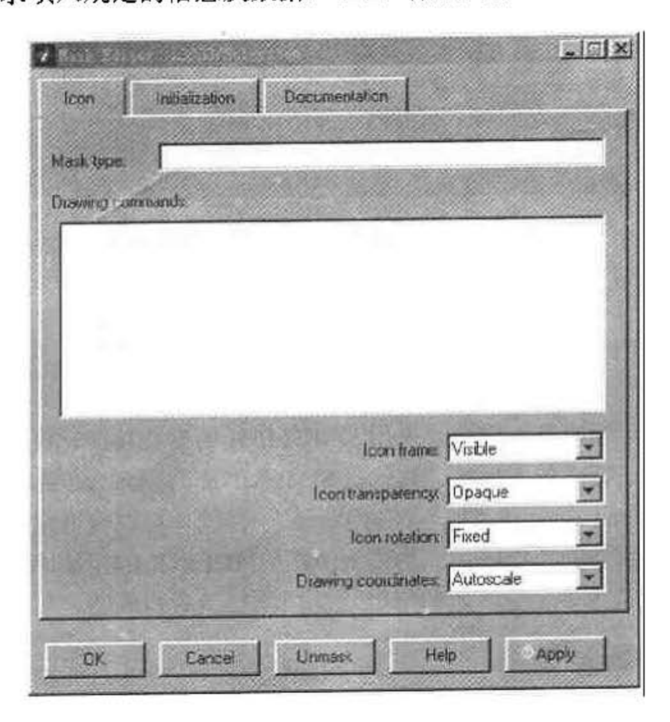


图 3-67　Mask 编辑器的 Icon 标签页

“封装”（Mask）过程操作步骤如下：首先，选取（中）要进行封装的子系统（Subsystem），在其模型窗口【Edit】菜单下选择【Mask Subsystem】命令。执行该命令，就会弹出“Mask Editor”（Mask 编辑器）。其次，按“Mask Editor”的三个标签页对话框的内容输入相关信息。

“Mask Editor”的三个标签页实际上是封装子系统成为特殊模块的三大类属性，它们是 Icon、Initialization、Documentation，其主要功能分别为：

（1）Icon —— 设定被封装子系统为特殊模块的图标外观。

（2）Initialization —— 设定参数的提示符（Prompt）、变量名（Variable）并指定初始化命令（Initialzation Commands）。

（3）Documentation —— 设计被封装子系统描述与帮助文本的文字说明。

3.9.3 Mask 的 Icon 标签页

“Icon” 标签页（图 3-67）能够为封装子系统的特殊模块制做一个新图标，并可以

在模块的图标上，以最能表示子系统封装模块功能特点的“文字说明”、“图像”以及“传递函数”等等信息标示在图标上。

在“Icon” 标签页对话框里有两个小编辑框和四个下拉式弹出菜单框，它们的标题是：“Mask type”、“Drawing commands”、“Icon frame”、“Icon transparency”、“Icon rotation”、“Drawing coordinates”。

在“Icon”标签页与其他的标签页对话框的下方还有 5 个按钮，它们是：【OK】、【Cancel】、【Unmask】、【Help】、【Applay】。除开【Unmask】按钮之外（【Unmask】按钮功能将在本节最后介绍），其他 4 个按钮已经介绍过，这里不再重复。

三个标签页都有“Mask type”栏，在栏右的编辑框里输入被封装子系统模块的类型。它可以是任意的字符串，这个被封装子系统的类型无任何实际意义。

在“Drawing coordinates”栏的编辑框里输入特定的命令能够建立封装子系统模块的图标，并可以在图标上显示文本、图像、图形或传递函数，其常用命令如下。

首先说明，以下都是在“Icon” 标签页编辑框“Drawing commands”内输入的命令。

（1）在图标上标注文字文本的命令如下：

disp（'string'）

命令里的 string 就是要显示在模块图标上所标注的文字字符串。当输入与 string 相应的命令时，执行后就可以在子系统模块图标上看到所加的文字字符串。

如果要选定文字在图标上的位置，则需使用命令：

text（x, y, 'string'）

命令中用（x, y）设定文字的起始位置。

（2）在图标上绘制图形的命令如下：

plot（[x1,x2,⋯, xn],[y1,y2,⋯, yn]）

该命令执行后，就会在图标上绘制出图形，图形是从 [x1, y1] 点经 [x2, y2] 点再经 [x3, y3]点……直到 [xn，yn]点为止所有两点间的直线。封装模块图标的左下角会根据目前的坐标刻度被正规化为 [0，0]，其右上角会根据目前的坐标刻度被正规化为 [1，1]。值得注意的是，x 和 y 的下标数目必须要相同，否则无法画出图形。

（3）在图标上标注传递函数的命令如下：

dpoly（***num, den***）

命令里的（***num, den***） 就是要标注的被封装子系统传递函数表达式的分子与分母多项式，分子与分母多项式是用其‘s’各阶导数按降幂排列的系数来表示的。

如果要标注零极点增益模型的系统传递函数时，可以用以下格式的命令：

Droots （*z, p, k*）

命令里的 *z,p*,k 分别是系统的零点、极点与增益。

如果要标注离散系统脉冲传递函数时，命令中还要多加入一个参数‘z’，命令格式为：

dpoly（***num, den***, 'z'）

（4）在图标上显示图像的命令如下：

image （ **imraed** 'imagename' ）

命令里的 imagename 是图像的文件名。

（5）封装子系统模块属性的确定

在“Icon”标签页的下部位置有四个弹出式菜单框，它们是：“Icon frame”、“Icon transparency”、“Icon rotation”、“Drawing coordinates”。它们用来对封装字系统模块图标的外观进行设计。

Icon frame：用来确定图标边框线条是否显示。“Visible”为显示边框，“Invisible”为隐藏边框。

Icon transparency：用来确定图标里的端口‘port’西文标记是否显示。“Opaque”为隐藏端口“in”与“out”的标记，“Transparent”为显示端口的“in”与“out”标记。

Icon rotation：用来确定图标是否旋转。选项“Fixed”表示图标不随被封装子系统的旋转而旋转，而“Rotates”则旋转。

Drawing coordinates：用来对画图时的坐标系进行设置，这个下拉式列表提供三种选择。“Autoscale”为自动选取坐标系并自动标刻度，坐标中最小的 x 与 y 位于图标左下角，最大的 x 与 y 位于图标右上角，当模块大小改变时图标大小随之改变，这是使用最多的一种选择。“Pixel”是以像素为单位绘制图形，当模块大小改变时图标大小不改变。

“Normalized”规定图标左下角坐标为（0，0），右上角的坐标为（1，1），要绘制图形的点必须归化到[0 1]之间才能显示出来。后两种选择很少被采用。

3.9.4 Mask 的 Initialization 标签页

Mask 的 Initialization 标签页（图 3-68）主要用于设计输入提示（Prompt）以及对应的变量名称（Variable）并指定初始化命令。

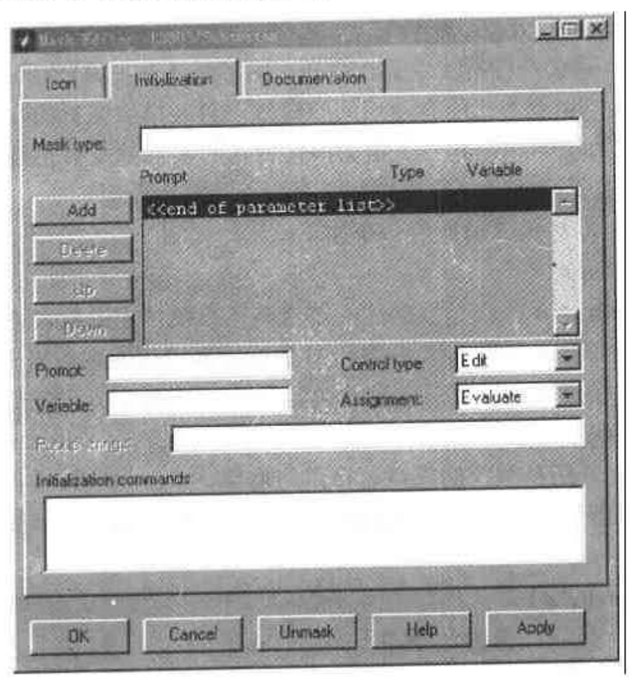


图 3-68　Mask 的 Initialization 标签页

在“Variable”栏里输入程序控制要用到的变量，该变量的值一直存储于 Mask workspace 中，因此可以与其他程序相互传递（参见以下举例）。输入的变量名应与在“Prompt”栏里输入的变量含义相对应（参见以下举例）。

在“Prompt”栏里输入变量的含义，其内容会显示在这个标签页中间部位的一个列表提示框中。如果要增加输入变量的话，可以用鼠标左键单击【Add】，会出现与刚才一样的输入框以供用户输入新的变量及其含义。请注意，最后要用按钮【Apply】命令对输入的信息予以确认。【Up】与【Down】按钮则用于列表窗口中执行变量间的顺序位置调整。当选定某一参数后，按下按钮【Up】为向上调，按【Down】为向下调。除可以增加变量用【Add】按钮外，还可以删除变量，则以鼠标左键单击【Delete】按钮即可。

若是设置完的变量与提示符不合要求，可以进行修改。其方法是：在列表框中选择欲修改变量所在的行，用鼠标左键点击之，此变量及其含义会分别显示在“Variable”与“Prompt”栏的编辑框里即可进行修改。变量间的上下顺序位置可用【Up】与【Down】按钮进行调节。对不合要求的设置可以重复进行多次修改，直到满意为止。

在“Assignmment”栏里有两项选择：“Evaluate”与“Literal”。一般使用情况下，选“Evaluate”时，输入的内容先由 MATLAB 进行计算，然后把结果赋予相关变量；选“Literal”时，输入的内容不进行计算，而是以字符串格式直接赋予相关变量。缺省设置为“Evaluate”。

在“Control type”栏里确定被封装子系统参数设置的控制方式，有三种可供选择的类型：“Edit”、“Checkbox”、“Popup”。

“Assignmment”栏与“Control type”栏配合使用的不同规则，参见表 3-9。

表 3-9 Assignment 不同参数的含义

	Evaluate	Literal
Edit	输入的内容经 MATLAB 程序计算结果赋予相关变量	输入的内容当作字符串赋予相关变量
Popup	返回值为菜单选项的序号，选第一项，则返回 1，往下依此类推	返回值为选项的字符串内容
Checkbox	返回值为 1 或 0	返回值为 ‘on’ 或 ‘off’

3.9.5 Mask 的 Documentation 标签页

在 Documentation 标签页（图 3-69）里，设计者可以针对设计完成的封装子系统模块来编写相应的 Help 帮助与说明文件。在‘Block description’对话框中输入被封装子系统的描述性文本文字后，会出现在属性窗口说明部分，而在‘Block help’中输入的帮助性文本文字则会显示在单击属性窗口中的【help】按钮后浏览器所加载的 HTML 文件中。

3.9.6 Mask 举例

【例 3-16】双闭环调速系统（图 3-70）有电流环子系统功能模块“Subsystem”（图 3-71），①试对电流环子系统进行“Mask”（封装）操作；②对封装后的双环系统进行仿真。

Icon | Initialization | Documentation

Mask type:

Block description:

Block help:

OK | Cancel | Unmask | Help | Apply

图 3-69　Mask 的 Documentation 标签页

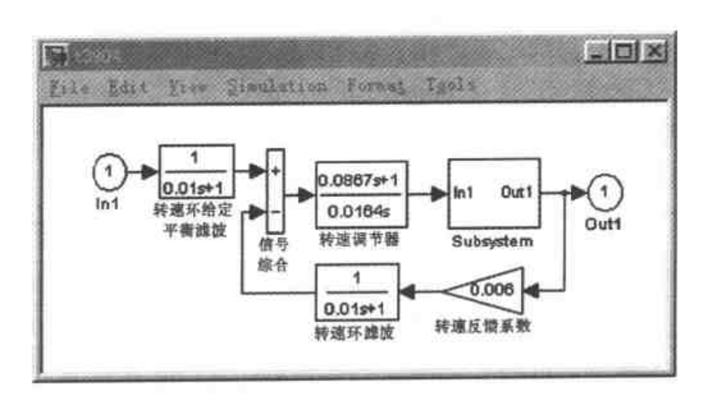


图 3-70　双环系统及其电流环子系统“Subsystem”

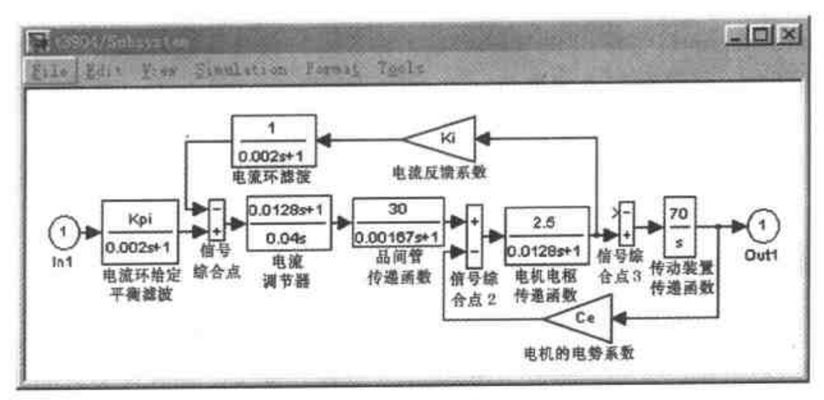


图 3-71　被封装的电流环子系统“Subsystem”

【解】

（1）Icon 标签页设置

在标签页里的“Mask type”栏里输入封装子系统模块的名称“Current loop Mask”。在“Drawing commands”框内输入以下命令（图 3-72）：

```
plot （[2.8 1 3.5 4.9 6.3 8.8 7 2.8],[1 5 5 7 5 5 1 1]）
text（3.5,6,'电流环传递函数'）
dpoly （135,[0.00000334 0.00367 1 135]）
```

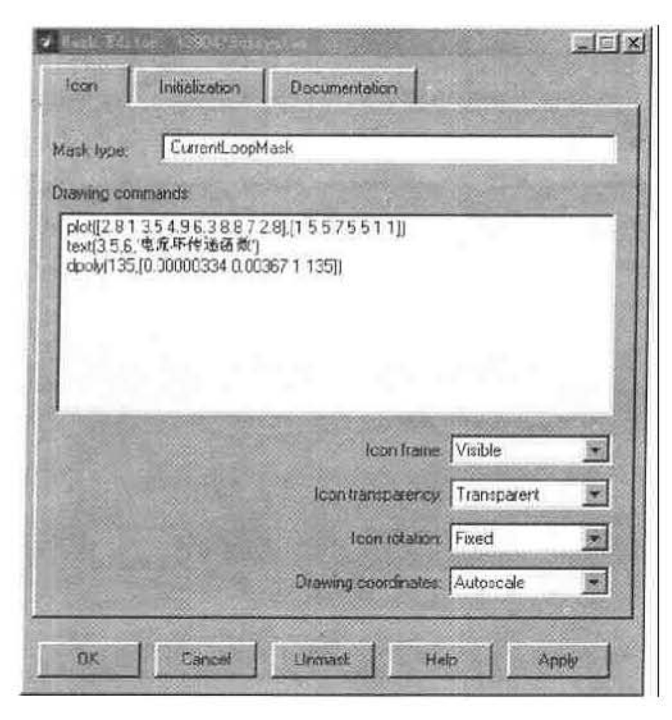


图 3-72 Icon 标签页设置图标

在 Icon transparency 栏选择“Transparent”选项，用以显示被封装子系统模块端口的“in”与“out”标记。在“Icon”标签页对话框下部按下【OK】按钮，就可以在封装子系统模块图标上看到所标示的文字说明、所画帽形图标及传递函数表达式（图 3-73）。

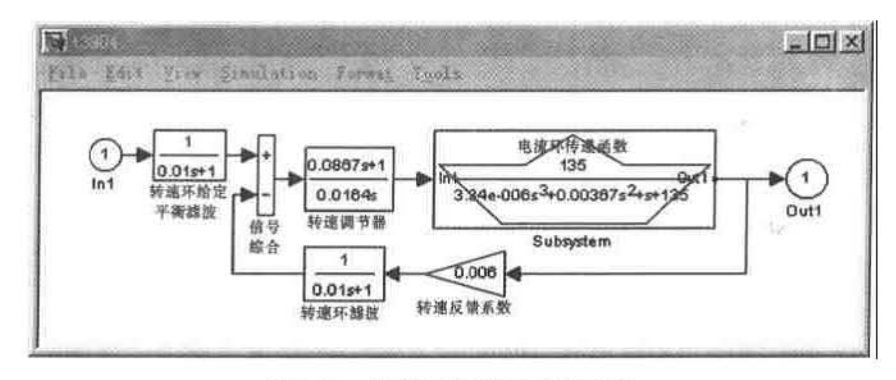


图 3-73 封装电流环子系统的图标

（2）Initialization 标签页设置

在标签页的“Variable”栏里输入变量的名称“Ki”，在“Prompt”栏里输入变量“Ki”的含义“电流反馈系数”，其内容会显示在 Initialization 标签页中间部位的一个列表提示框中（参见图 3-74）。如果要增加变量的话，可以用鼠标左键单击【Add】按钮，会出现与刚才一样的输入框以供用户输入新的变量。如果在“Variable”栏里再分别输入名称为“Ce”、“Kpi”的变量，在‘Prompt’栏里再分别输入其变量的含义“电机电势系数”、“电流环的增益”，则出现如图 3-74 所示的界面。还可以用【Up】与【Down】按钮将列表窗口中执行变量间的位置调整为所需要的情况。最后，再以鼠标左键单击【Apply】按钮，此意为对输入的变量及其含义进行确认，即表示将目前窗口改变的参数设定应用于系统以下的仿真，并保持对话框窗口的开启状态，以便进一步修改。设置完后显示的画面如图 3-74 所示。

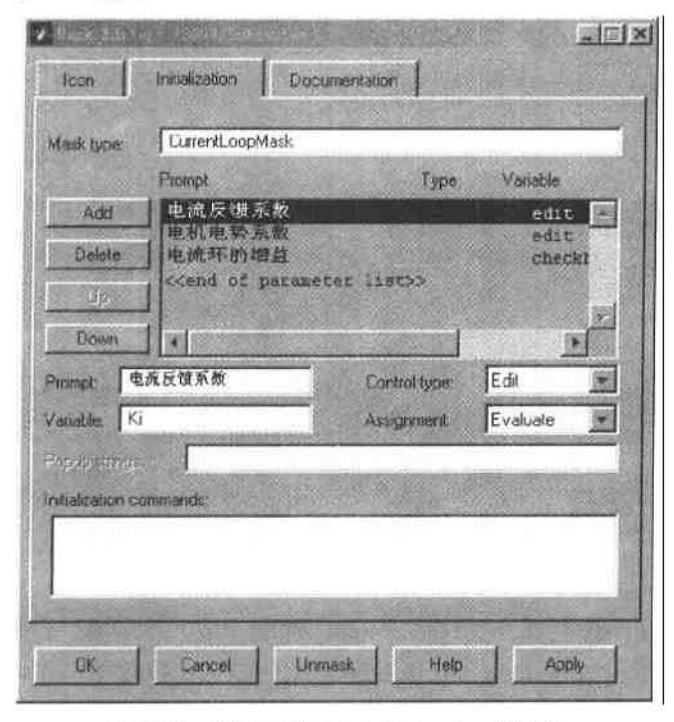


图 3-74　子系统封装的 Initialization 标签页

参数设置完后，在“Control type”栏里选择“Edit”选项，按下 Initialization 标签页底部的【OK】按钮。再回到模型窗口（图 3-73）中用鼠标左键双击子系统（Subsystem）图标，则会弹出子系统模块参数设置对话框，如图 3-75 所示，就可以在参数设置对话框中输入或修改参数了。

需要再次指出，参数必须分别输入在“电流反馈系数”、“电机电势系数”、“电流环的增益”等栏下部的编辑框内，并且只输入变量的数值或表达式，不能带变量名与等号（参见图 3-75）。

在“Control type”栏里选择“Popup”与“Checkbox”选项，可以做类似的操作。

CurrentLoopMask (mask)

Parameters

电机电势系数

0.136

电流环的增益

3

电流反馈系数

0.072

OK　Cancel　Help　Apply

图 3-75　Initialization 标签的 Edit 选项时设置参数

（3）Documentation 标签页设置

在 Documentation 标签页里，设计被封装子系统描述文本与帮助文本的文字说明。在“Block description”栏里输入以下描述文本“电流环是直流双闭环调速系统的内环，包括电流调节器、可控硅移相触发与整流、电机及其传动装置，电流环内有交叉反馈。”，在“Block help”栏里输入以下帮助文本“Ki=0.072 V/A，Ce=0.136V/（r/min），当 Kpi=1 与 Kpi=3 两种情况下的阶跃给定响应不同，前者不振荡，后者衰减振荡。”然后按下【OK】按钮，则 Documentation 标签页设置完成，参看图 3-76。

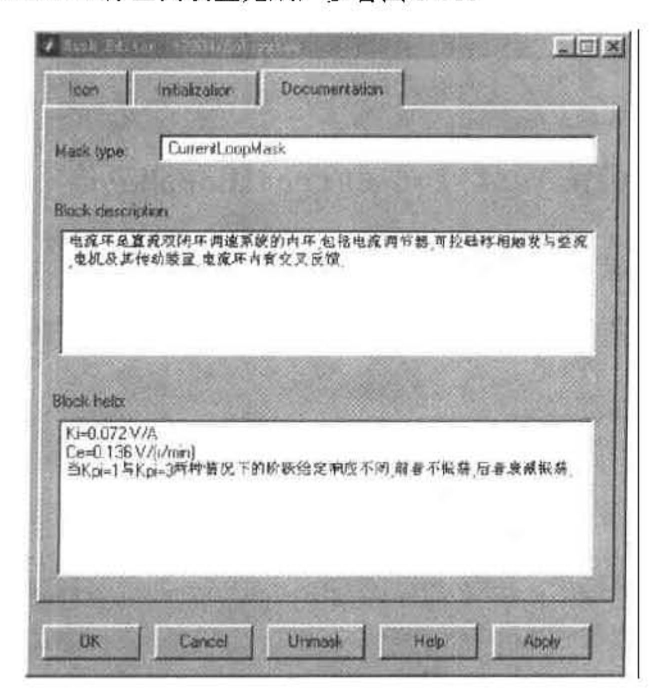


图 3-76　电流环子系统封装 Documentation 标签

再回到模型窗口（图 3-73）中用鼠标左键双击被封装子系统图标，则会弹出子系统模块带描述文本的对话框，如图 3-77 所示，可以看出，图 3-77 就是图 3-75 在“CurrentLoopMask （mask）”栏下添加了描述文本的内容，两图其他的内容相同。

CurrentLoopMask (mask)
电流环是直流双闭环调速系统的内环，包括电流调节器，可控硅移相触发与整流，电机及其传动装置，电流环内有交叉反馈。
Parameters
电机电势系数
0.136
电流环的增益
3
电流反馈系数
0.072
OK　Cancel　Help　Apply

图 3-77　封装 Documentation 标签的描述说明

在图 3-77 的底部按下【Help】按钮，即可看到图 3-78 所示的被封装子系统帮助说明，其文本标题是“Help Text for CurrentLoopMask”，其内容正是在“Block help”栏里输入的帮助文本（图 3-76）。图 3-78 中还有有关 MATLAB 系统其他内容与 SIMULINK 的帮助说明，这只需要用鼠标左键双击“Help Desk”部分或者双击“Simulink”部分即可。图 3-78 所示的窗口实际上就是 MATLAB 的帮助窗口。

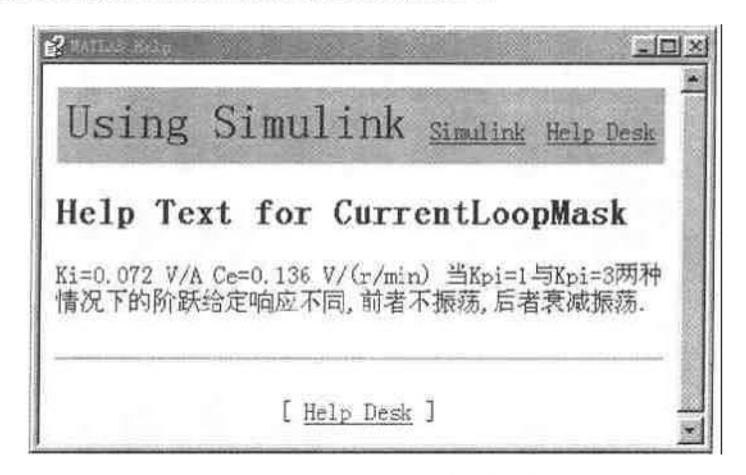


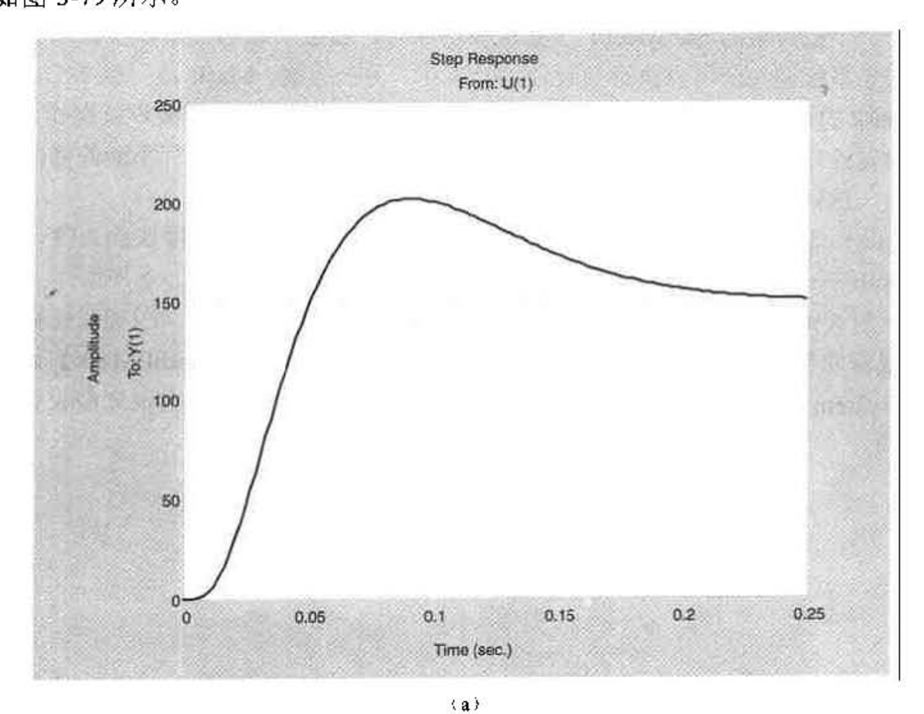
图 3-78　封装 Documentation 标签的帮助说明

（4）封装后系统仿真

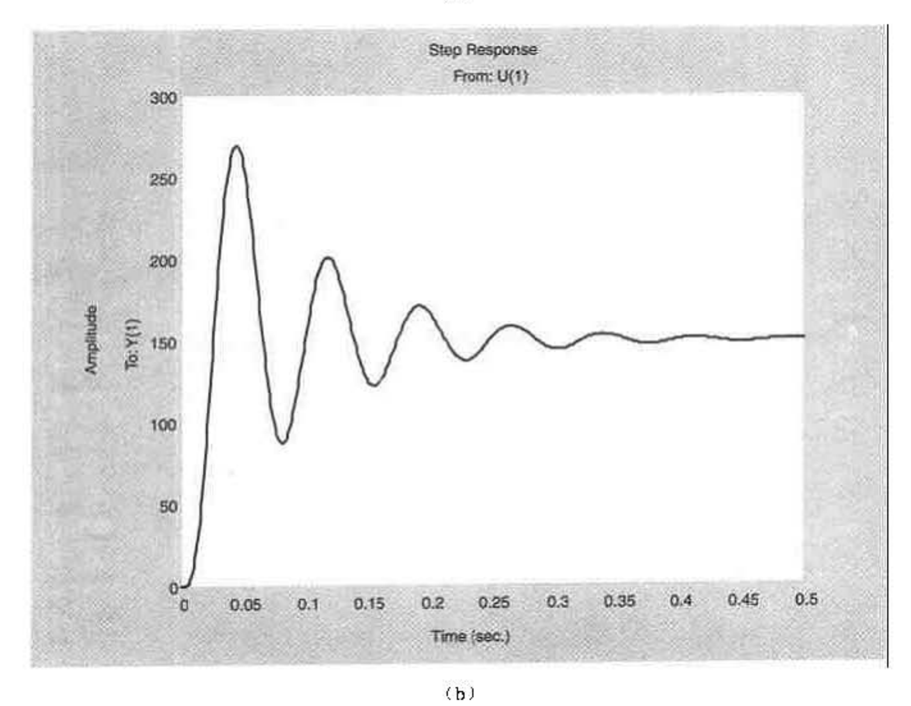
封装了系统“Subsystem”内有三个变量“*Ki*”、“*Ce*”、“*Kpi*”，通过封装及 Initialization 标签的设定，存放在 Mask workspace 空间中的信息数据可以传递给程序来运算执行。以下给出被封装子系统的双闭环调速系统的阶跃给定响应仿真程序 L3901.m。

```
%MATLAB PROGRAM L3901.m
%
cd c:\matlabr11\work
[A,B,C,D]= linmod ('t3904');
step (A,B,C,D)
```

当设置 *Ki*=1 与 *Ki*=3 两个值时，程序运行后，可以分别得到系统的阶跃给定响应曲线，如图 3-79 所示。



（a）



（b）

图 3-79　封装后双环系统的阶跃给定响应曲线

（a）Kpi=1；（b）Kpi=3。

3.9.7 Mask 的 Unmask 功能

对于一个被封装的 Subsystem 子系统模块而言，通过上述的 Mask 步骤之后，该集成封装的子系统模块就可以执行所设计的功能了。另一方面，Mask 还提供了 Unmask 功能。Unmask 的作用是对于在封装过程中的子系统撤销最近一次的封装设置操作，回到最近一次设置操作之前的状态。这样即可通过 Unmask 对封装模块进行不断的修改，直到满意为止。这样完善的功能会让用户总能够设计出理想的系统来。

Unmask 具体操作如下：首先，选择待封装 Subsystem 子系统模块的系统，鼠标左键双击 Subsystem 图标；其次，选择【Edit】菜单下的【Edit Mask】命令并执行。在 Mask Editor（封装编辑器）里，鼠标左键单击【Unmask】按钮；最后，回到系统模型窗口里，再鼠标左键双击 Subsystem 图标，在【Edit】菜单下发现【Edit Mask】已经变成【Msk System】了，选择【Msk System】选项命令，执行后就会出现先前设计的系统以供修改。

第 4 章　控制系统 MATLAB 仿真基础

本章主要介绍控制系统的数学模型，控制系统的稳定性分析、稳态误差分析，控制系统计算与仿真的基本方法等内容。本章共 6 节：4.1 节介绍自动控制系统数学模型的建立以及模型之间的转换；4.2 节介绍控制系统稳定性分析的 MATLAB 实现；4.3 节介绍控制系统稳态误差分析的 MATLAB 实现；4.4 节介绍时域响应的 MATLAB 仿真；4.5 节介绍频域响应的 MATLAB 仿真；4.6 节介绍系统根轨迹分析的 MATLAB 实现。通过本章的学习，可以对系统的数学模型、各种仿真的主要问题、内容与方法有最基本的了解，为控制系统的 MATLAB 仿真打下基础。本章不可能复习与自动控制相关的全部内容，但是作为自动控制的主要基本概念还是有必要作简单的复习，这在某些章节的开始部分作了相应安排。在 4.3 节与 4.4 节后附有作者开发的三个 MATLAB 函数。

4.1　控制系统的数学模型

4.1.1　LTI 对象

线性定常时不变（LTI）对象有三类：ss 对象、tf 对象与 zpk 对象，它们分别为状态空间模型、传递函数模型与零极点增益模型。每个对象都有其对象属性与对象方法，同类对象的属性可以继承，通过对象方法可以存取或者设置对象属性值。三类对象有共同的属性，也有各自的特有属性。其共同属性参见表 4-1。

表 4-1　模型共有属性表

属 性 名	功　能	属性值类型
Ts	采样周期	数值标量
InputDelay	输入延时	数值向量
OutputDelay	输出延时	数值向量
ioDelayMatrix	输入输出延时	数值数组
InputName	输入变量名	字符串单元向量
OutputName	输出变量名	字符串单元向
InputGroup	输入通道组	M 输入组单元向量
OutputGroup	输出通道组	P 输出组单元向量
Notes	模型描述	文本
Userdata	附加数据	任意数据类型

表中采样周期 *Ts* 既可用于连续系统，也可用于离散系统。对离散系统，其属性值给出了系统的采样周期。对连续系统，*Ts*=0。当 *Ts*= -1 时，表示离散系统采样周期未定。输入延时仅对于连续系统有效，其值为每个输入通道的输入延时组成的时间向量，输入延时=0 为缺省值，表示无输入延时。

输入变量名 **InputName** 与输出变量名 **OutputName** 允许用户定义系统输入输出的名称。其值为一个字符串单元向量，分别与输入输出具有相同的维数，缺省值为空。

输入通道组 InputGroup 与输出通道组 OutputGroup 的属性值都是单元向量。

模型描述 Notes 与附加数据 Userdata 用于存储模型的其他信息，前者用于给出模型描述的文本，后者可用于存储用户需要的任意其他数据，缺省值为空。

ss 对象特有属性参见表 4-2。set(ss)函数指令可定义状态空间模型对象的所有属性：

```
                a: Nx-by-Nx matrix (Nx = no. of states)
                b: Nx-by-Nu matrix (Nu = no. of inputs)
                c: Ny-by-Nx matrix (Ny = no. of outputs)
                d: Ny-by-Nu matrix
                e: Nx-by-Nx matrix (or [])
        StateName: Nx-by-1 cell array of strings
               Ts: scalar
       InputDelay: Nu-by-1 vector
      OutputDelay: Ny-by-1 vector
    ioDelayMatrix: Ny-by-Nu array (I/O delays)
        InputName: Nu-by-1 cell array of strings
       OutputName: Ny-by-1 cell array of strings
       InputGroup: M-by-2 cell array if M input groups
      OutputGroup: P-by-2 cell array if P output groups
            Notes: array or cell array of strings
         UserData: arbitrary
Type "ltiprops ss" for more details.
```

表 4-2　ss 对象特有属性表

属　性　名	功　　能	属性值类型
a	系数矩阵 ***A***	二维矩阵
b	系数矩阵 ***B***	二维矩阵
c	系数矩阵 ***C***	二维矩阵
d	系数矩阵 ***D***	二维矩阵
e	系数矩阵 ***E***	二维矩阵
StateName	状态变量名	字符串单元向量

类似于 set (ss) 函数命令执行的结果，函数 set (tf) 可以定义传递函数模型对象的所有属性，tf 模型对象特有属性参见表 4-3；函数 set(zpk)可以定义零极点增益模型对象的所有属性，zpk 模型对象特有属性参见表 4-4。

表 4-3　tf 对象特有属性参见表

属 性 名	功　能	属性值类型
den	传递函数分母系数	由数值行向量组成的单元阵列
num	传递函数分子系数	由数值行向量组成的单元阵列
Variable	传递函数变量	's','p','z','q'或者'z^-1'之一

表 4-4　zpk 对象特有属性表

属 性 名	功　能	属性值类型
k	增益	二维矩阵
p	极点	由数值行向量组成的单元阵列
Variable	零极点模型变量	's','p','z','q'或者'z^-1'之一
z	零点	由数值行向量组成的单元阵列

ss 对象的 e 属性用来定义描述状态空间模型中的系数矩阵，在标准的状态空间模型中 e = ***I***。式中，***I*** 为单位矩阵。

ss 对象的 StateName 属性类似于 InputName 属性，它用来定义状态空间模型中每个状态的名称。

tf 对象与 zpk 对象的 Variable 属性用于定义传递函数的频率变量。缺省时，连续系统为 '*s*'（Laplace 算子 *s*）；离散时间系统为'*z*'（*z* 变换变量 *z*）；也可以选择 '*p*'、'*q*'或者'*z*^-1'。其主要作用是显示不同传递函数的形式。

4.1.2　控制系统数学模型的种类与转换

控制系统的数学模型通常是指动态数学模型。自动控制系统最基本最重要的数学模型是微分方程，它反映部件或系统动态运行的规律。应用拉普拉斯变换求解线性常系数微分方程时引出了一个传递函数的概念，以及基于传递函数的图形化形式——动态结构图，都是自动控制系统的数学模型。

实际工程里，要解决自动控制问题所需用的数学模型与该问题所给定的已知数学模型往往不一致；或者要解决问题最简单而又最方便的方法所用到的数学模型与该问题所给定的已知数学模型不同，此时，就需要对自控系统的数学模型进行转换。

在 MATLAB 里，可用 4 种数学模型表示控制系统，其中前 3 种是用数学表达式描述的：传递函数模型、零极点增益模型与状态空间模型。每种模型都有连续系统的及离散系统的两种类别。基于传递函数的图形化形式——动态结构图就是 MATLAB 里的 SIMULINK 结构图为第 4 种数学模型。

4.1.2.1　控制系统数学模型

MATLAB 5.3 里关于系统数学模型函数命令的功能比以前的版本有了很大的变化，建立某种数学模型的函数增添了模型转换功能。这样，用 MATLAB 函数编制的程序更加简单、精炼而高效。

1．传递函数模型

线性定常时不变对象（LTI）可以是连续时间系统，也可以是离散时间系统。连续时间 LTI 系统常以定常系数线性常微分方程来描述。对于一个连续单输入单输出 (SISO)的 LTI 系统，设系统输入量为 $r(t)$，输出量为 $c(t)$，则系统的微分方程为：

$$\begin{aligned} &a_1\frac{\mathrm{d}^n c(t)}{\mathrm{d}t^n}+a_2\frac{\mathrm{d}^{n-1}c(t)}{\mathrm{d}t^{n-1}}+\cdots+a_n\frac{\mathrm{d}c(t)}{\mathrm{d}t}+a_{n+1}c(t)= \\ &b_1\frac{\mathrm{d}^m r(t)}{\mathrm{d}t^m}+b_2\frac{\mathrm{d}^{m-1}r(t)}{\mathrm{d}t^{m-1}}+\cdots+b_m\frac{\mathrm{d}r(t)}{\mathrm{d}t}+b_{m+1}r(t) \end{aligned} \tag{4-1}$$

对应的传递函数定义为：在零初始条件下，系统输出量的拉氏变换 $C(s)$ 与输入量的拉氏变换 $R(s)$ 之比如下：

$$G(\mathrm{s})=\frac{C(s)}{R(s)}=\frac{b_1s^m+b_2s^{m-1}+\cdots+b_{m+1}}{a_1s^n+a_2s^{n-1}+\cdots+a_{n+1}}=\frac{\boldsymbol{num}(s)}{\boldsymbol{den}(s)} \tag{4-2}$$

对于离散时间系统，常以定常系数线性差分方程来描述。其单输入单输出 (SISO)的 LTI 系统的微分方程为：

$$\begin{aligned} &a_1c(k+n)+a_2c(k+n-1)+\cdots+a_nc(k+1)+a_{n+1}c(k)= \\ &b_1r(k+m+1)+b_2r(k+m)+\cdots+b_mr(k+1)+b_{m+1}r(k) \end{aligned} \tag{4-3}$$

对应的脉冲传递函数为：

$$G(\mathrm{z})=\frac{C(z)}{R(z)}=\frac{b_1z^m+b_2z^{m-1}+\cdots+b_{m+1}}{a_1z^n+a_2z^{n-1}+\cdots+a_{n+1}}=\frac{\boldsymbol{num}(z)}{\boldsymbol{den}(z)} \tag{4-4}$$

不论是连续还是离散时间系统，传递函数分子/分母均按‘s’或‘z’的降幂排列。在 MATLAB 里，都可直接用分子/分母多项式系数构成的两个向量 ***num*** 与 ***den*** 表示系统，即

$$\begin{cases} \boldsymbol{num}=[b_1,b_2,\cdots,b_m] \\ \boldsymbol{den}=[a_1,a_2,\cdots,a_n] \end{cases} \tag{4-5}$$

在 MATLAB 中，用函数命令 tf()来建立控制系统的传递函数模型，或者将零极点模型或者状态空间模型转换为传递函数模型。tf( )函数命令的调用格式为：

sys=**tf(*num*,*den*)**

sys=**tf**(*num*,*den*,*Ts*)

sys=**tf**(*M*)

sys=**tf(*num*,*den***,*ltisys*)

sys=**tf(*num*,*den***,'*Property*1'，*Value*1，…，'*PropertyN*'，*ValueN*)

sys=**tf(*num*,*den***,*Ts*，'*Property*1'，*Value*1，…，'*PropertyN*'，*ValueN*)

tfsys=**tf**(*sys*)

tfsys=**tf**(*sys*，'*inv*')

sys=tf(***num*,*den***)函数返回的变量 *sys* 为连续系统的传递函数模型。函数输入参量 ***num*** 与 ***den*** 分别为系统的分子与分母多项式系数向量。

sys=tf(***num*,*den***, *Ts*)函数返回的变量 *sys* 为离散系统的传递函数模型。函数输入参量 ***num*** 与 ***den*** 分别为系统的分子与分母多项式系数向量，*Ts* 为采样周期，当 *Ts*=-1 或者 *Ts*=[] 时，则系统的采样周期未定义。

sys=tf(*M*) 函数定义一个增益为M的静态系统。

sys=tf(***num*,*den***,*ltisys*)函数创建的 tf 对象的其他属性将继承 *ltisys* 的属性，包括采样周期。

sys=tf(***num*,*den***,*'Property1'*, *Value1*,…,*'PropertyN'*,*ValueN*)与该函数带 *Ts* 的函数创建 tf 对象并定义其他属性值。其中 *Property1* 与 *PropertyN* 为属性标识字符串，*Value1* 与 *ValueN* 为对应的属性值。

tfsys=**tf**(*sys*)函数将任意的 LTI 对象转换成传递函数模型，缺省时使用 tzero ( ) 函数将状态空间模型转换成传递函数模型，使用 poly ( ) 函数将零极点增益模型转换成传递函数模型。tzero ( ) 函数用来计算系统传输零点，poly ( ) 函数已在 2.3 节中介绍过，其功能是由多项式的根求多项式。pole ( )函数用来计算系统极点。如果需要这些函数的详细信息，可以用 help 命令获得帮助。

tfsys=**tf**(*sys*,*'inv'*)函数使用函数 pole ( ) 将状态空间模型转换成传递函数模型。

离散系统的脉冲传递函数的表达式还有一种表示为 z^{-1} 的形式（即 DSP 形式），转换为 DSP 形式脉冲传递函数的函数命令为 filt()。filt()函数调用格式为：

sys=**filt**(***num*,*den***)

sys=**filt**(***num*,*den***,*Ts*)

sys=**filt**(*M*)

sys=**filt**(***num*,*den***) 函数用来建立一个采样时间未指定的 DSP 形式脉冲传递函数。函数输入参量 ***num*** 与 ***den*** 分别为系统的分子与分母多项式系数向量。

sys=**filt**(***num*,*den***,*Ts*)函数用来建立一个采样时间由 *Ts* 指定的 DSP 形式脉冲传递函数。函数输入参量 ***num*** 与 ***den*** 分别为系统的分子与分母多项式系数向量。

sys=**filt**(*M*)函数定义一个增益为M的静态系统。

需要指出，对于已知的传递函数，其分子与分母多项式系数向量可分别由 G.***num***{1} 与 G.den{1}指令求出。这种指令对于程序设计是非常有用的。

传递函数是线性定常系统的一种重要数学模型。它不仅可以表征系统的动态特性，而且还可用来研究系统的结构或参数变化对系统性能的影响。

在 MATLAB 中，还用函数 printsys ( )来输出控制系统的传递函数，printsys ( )函数的调用格式为：

printsys(***num*,*den***,*'s'*)

printsys(***num*,*den***,*'z'*)

式中 ***num*** 与 ***den*** 分别为系统的分子与分母多项式系数向量，*'s'*是指对连续系统的拉氏变换，输出的是连续系统的传递函数模型。*'z'*是指对离散系统的 *z* 变换，输出的是离散系统的脉冲传递函数模型。

2. 零极点增益模型

连续系统传递函数表达式用系统增益、系统零点与系统极点来表示的，则叫做系统零极点增益模型。可以说系统零极点增益模型是传递函数模型的一种特殊形式。即有

$$G(s)=k\frac{(s+z_1)(s+z_2)\cdots(s+z_m)}{(s+p_1)(s+p_2)\cdots(s+p_n)} \tag{4-6}$$

离散系统传递函数也可用系统增益、系统零点与系统极点来表示：

$$G(z)=k\frac{(z+z_1)(z+z_2)\cdots(z+z_m)}{(z+p_1)(z+p_2)\cdots(z+p_n)} \tag{4-7}$$

式中，k 为系统增益，z_1、z_2、z_m 为系统零点，p_1、p_2、p_m 为系统极点。在 MATLAB 里，连续与离散系统都可直接用向量 $\boldsymbol{z}$、$\boldsymbol{p}$、$\boldsymbol{k}$ 构成的矢量组$[\boldsymbol{z},\boldsymbol{p},\boldsymbol{k}]$表示系统，即

$$\begin{cases}\boldsymbol{z}=[z_1,z_2,\cdots,z_m]\\ \boldsymbol{p}=[p_1,p_2,\cdots,p_n]\\ \boldsymbol{k}=[k]\end{cases} \tag{4-8}$$

在 MATLAB 中，用函数命令 zpk( )来建立控制系统的零极点增益模型，或者将传递函数模型或者状态空间模型转换为零极点增益模型。zpk( )函数的调用格式为：

sys=**zpk(*num*,*den*)**

sys=**zpk(*num*,*den*,*Ts*)**

sys=**zpk**(*M*)

sys=**zpk(*num*,*den*,*ltisys*)**

sys=**zpk(*num*,*den*,*'Property1'*,*Value1*,…,*'PropertyN'*,*ValueN*)**

sys=**zpk(*num*,*den*,*Ts*,*'Property1'*,*Value1*,…,*'PropertyN'*,*ValueN*)**

tfsys=**zpk**(*sys*)

tfsys=**zpk**(*sys*,*'inv'*)

sys=**zpk(*num*,*den*)**函数返回的变量 *sys* 为连续系统的零极点增益模型。函数输入参量的含义同 tf( )函数命令的解释。需要指出，对于已知的零极点增益模型传递函数，其零点与极点可分别由 sys.z{1}与 sys.p{1}指令求出。这也是一个非常有用的指令，它对于编制 MATLAB 程序也会带来很大方便。

3．状态空间模型（详见 7.1 节）

控制系统在主要工作区域内的一定条件下可近似为线性时不变（LTI）模型，连续 LTI 对象系统总是能用一阶微分方程组来表示，写成矩阵形式即为状态空间模型：

$$\begin{cases}\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\,\boldsymbol{x}(t)-\boldsymbol{B}\,\boldsymbol{u}(t) & (a)\\ \boldsymbol{y}(t)=\boldsymbol{C}\,\boldsymbol{x}(t)+\boldsymbol{D}\,\boldsymbol{u}(t) & (b)\end{cases} \tag{4-9}$$

其中，式（4-9）(*a*)由 n 个一阶微分方程组成，叫做状态方程，式（4-9）(*b*)由 1 个线性代数方程组成，叫做输出方程。式中 $\boldsymbol{u}(t)$ 是 $r\times1$ 的系统控制输入(r 个)向量，$\boldsymbol{x}(t)$ 是 $n\times1$ 的系统状态变量，$\boldsymbol{y}(t)$ 则是 $m\times1$ 的系统输出向量。$\boldsymbol{A}$ 为 $n\times n$ 的系统矩阵（或称状态矩阵），由控制对象的参数决定；$\boldsymbol{B}$ 为 $n\times r$ 的控制矩阵（或称输入矩阵）；$\boldsymbol{C}$ 为 $m\times n$ 的输出矩阵（或称观测矩阵）；$\boldsymbol{D}$ 为 $m\times r$ 的输入输出矩阵（或称直接传输矩阵）。

离散系统的状态空间模型为：

$$\begin{cases}\boldsymbol{x}(k+1)=\boldsymbol{A}\,\boldsymbol{x}(k)+\boldsymbol{B}\,\boldsymbol{u}(k)\\ \boldsymbol{y}(k)=\boldsymbol{C}\,\boldsymbol{x}(k)+\boldsymbol{D}\,\boldsymbol{u}(k)\end{cases} \tag{4-10}$$

式中 $\boldsymbol{u}$, $\boldsymbol{x}$, $\boldsymbol{y}$ 分别为控制输入向量、状态向量、输出向量，k 表示采样点。$\boldsymbol{A}$ 为状态矩阵，

由控制对象的参数决定；**B** 为控制矩阵；**C** 为输出矩阵；**D** 为直接传输矩阵。

在 MATLAB 里，连续与离散系统都可直接用矩阵组 [**A**,**B**,**C**,**D**] 表示系统。

在 MATLAB 中，用函数 ss()来建立控制系统的状态空间模型，或者将传递函数模型与零极点增益模型转换为系统状态空间模型。ss( )函数的调用格式为：

sys=**ss**(*a*,*b*,*c*,*d*)

sys=**ss**(*a*,*b*,*c*,*d*,*Ts*)

sys=**ss**(*d*)

sys=**ss**(*a*,*b*,*c*,*d*,*ltisys*)

sys=**ss**(*a*,*b*,*c*,*d*,'*Property*1',*Value*1,...,'*PropertyN* ',*ValueN*)

sys=**ss**(*a*,*b*,*c*,*d*,*Ts*,'*Property*1',*Value*1,...,'*PropertyN*',*ValueN*)

*sys*_**ss**=**ss**(*sys*)

sys=ss(*a*,*b*,*c*,*d*)函数返回的变量 *sys* 为连续系统的状态空间模型。函数输入参量 *a*,*b*,*c*,*d* 分别对应于系统的 **A**,**B**,**C**,**D** 参数矩阵。

sys=ss(*a*,*b*,*c*,*d*,*Ts*)函数返回的变量 *sys* 为离散系统的状态空间模型。函数输入参量 *a*,*b*,*c*,*d* 分别对应于系统的 **A**,**B**,**C**,**D** 参数矩阵，*Ts* 为采样周期，当 *Ts*=-1 或者 *Ts*=[ ]时，则系统的采样周期未定义。

sys=ss(*d*) 函数等价于 *sys*=ss([ ],[ ],[ ], *d*)。

sys=ss(*a*,*b*,*c*,*d*,*ltisys*)函数创建的 tf 对象的其他属性将继承 ltisys 的属性，包括采样周期。

sys=ss(*a*,*b*,*c*,*d*,'*Property*1',*Value*1,...,'*PropertyN*',*ValueN*)与该函数带 *Ts* 的函数创建 tf 对象并定义其他属性值。其中 *Property*1 与 *PropertyN* 为属性标识字符串 Value1 与 ValueN 对应的属性值。

*sys*_*ss*=ss(*sys*)函数将任意的 LTI 对象 *sys* 转换成状态空间模型。

4.1.2.2 三种系统数学模型之间的转换

解决实际问题时，常常需要对自控系统的数学模型进行转换。前面介绍的系统数学模型建立函数本身有转换功能，但是在很多情况下使用这些函数还是不能适应解决问题的需要，或者很不方便。在 MATLAB 5.3 以前版本中的信号处理工具箱与控制系统工具箱中，提供了前三种数学表达式数学模型(传递函数模型、零极点增益模型与状态空间模型)之间转换的函数：ss2tf(), ss2zp(), tf2ss(), tf2zp(), zp2ss(), zp2tf()。这些函数之间的转换功能可用表 4-5 说明。

表 4-5　数学模型转换函数及其功能

函　数　名	函　数　功　能
ss2tf	将系统状态空间模型转换为传递函数模型
ss2zp	将系统状态空间模型转换为零极点增益模型
tf2ss	将系统传递函数模型转换为状态空间模型
tf2zp	将系统传递函数模型转换为零极点增益模型
zp2ss	将系统零极点增益模型转换为状态空间模型
zp2tf	将系统零极点增益模型转换为传递函数模型

这 6 个函数在某些情况下解决问题非它们莫属。

4.1.2.3 系统数学模型建立与转换举例

【例 4-1】两级 RC 阻容网络如图 4-1 所示，试求以 u_c 作输出，以 u_r 作输入的传递函数模型与零极点增益模型。

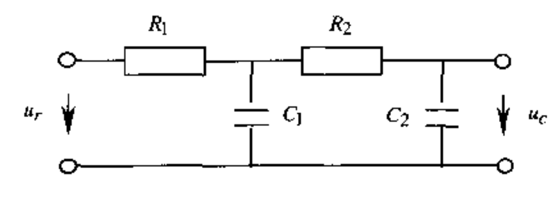


图 4-1　两级 RC 网络

【解】

系统传递函数为

$$\Phi(s)=\frac{U_c(s)}{U_r(s)}=\frac{1}{R_1C_1R_2C_2s^2+(R_1C_1+R_2C_2+R_1C_2)s+1}$$

即系统为：

$$\boldsymbol{num}=[0\ 0\ 1]$$

$$\boldsymbol{den}=[\,R_1C_1R_2C_2\quad (R_1C_1+R_2C_2+R_1C_2)\quad 1\,]$$

若　　$R_1=20\,\text{k}\Omega, R_2=40\,\text{k}\Omega, C_1=50\,\mu\text{F}, C_2=25\,\mu\text{F}$

$$\boldsymbol{den}=[\,R_1C_1R_2C_2\quad (R_1C_1+R_2C_2+R_1C_2)\quad 1\,]=[1\ 2.5\ 1]$$

代入数据后有 MATLAB 语句：

```
num=[1];den=[1 2.5 1];sys1=tf(num,den);printsys(num,den,'s')
sys=zpk(sys1)
```

执行以上语句可得系统传递函数模型与零极点增益模型为：

```
          1
num/den=---------------
         s^2 + 2.5 s + 1
Zero/pole/gain:
      1
-------------
(s+2) (s+0.5)
```

【例 4-2】控制系统校正时常采用的比例积分（PI）调节器（图 4-2）以 U_{ex} 作输出量，以 U_{in} 作输入量。试求同矢量组$[\boldsymbol{z},\boldsymbol{p},\boldsymbol{k}]$表示的系统零极点增益模型与传递函数模型。

【解】

系统传递函数为

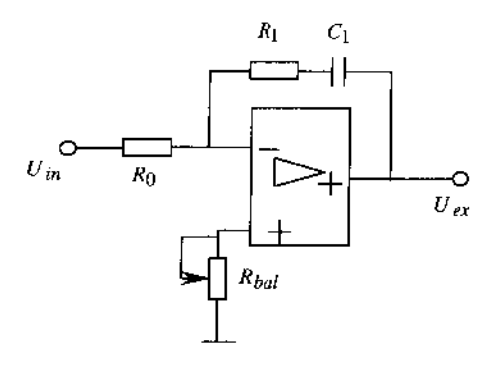


图 4-2　比例积分（PI）调节器

$$\Phi(s)=\frac{U_{ex}(s)}{U_{in}(s)}=\frac{K_{pi}\left(s+\frac{1}{K_{pi}\tau}\right)}{s}$$

式中：

$$K_{pi}=\frac{R_1}{R_0},\ \tau=R_0C_1$$

即系统为：

$$z=[-\frac{1}{K_{pi}\tau}];\quad p=[0];\ k=K_{pi}$$

若　$R_0=40\ \text{k}\Omega, R_1=1200\ \text{k}\Omega,\ C_1=5\,\mu\text{F}$，则

$$k=K_{pi}=\frac{R_1}{R_0}=1200/40=30;\quad \tau=R_0\cdot C_1=2\ \text{s};\ z=[-\frac{1}{K_{pi}\tau}]=-0.0167;\ p=[0]$$

为求系统的零极点增益模型与传递函数模型，给出以下语句：

```
k=30;z=-0.0167;p=0;sys=zpk(z,p,k)
sys1=tf(sys)
```

执行以上语句可得系统零极点增益模型与传递函数模型为：

```
Zero/pole/gain:
30 (s+0.0167)
-------------

      s
Transfer function:
30 s + 0.501
------------

     s
```

【例 4-3】已知一造纸工业中的网前箱为一加压液流箱系统。该系统的状态变量是箱中的液位 $h(t)$ 与料浆的总压头 $H(t)$，输入变量是料浆流入量 $u_1(t)$ 与空气流入量 $u_2(t)$，输出变量就是状态变量 $H(t)$ 与 $h(t)$ 本身。系统状态空间模型为：

$$\begin{cases}\begin{bmatrix}\dot{H}(t)\\ \dot{h}(t)\end{bmatrix}=\begin{bmatrix}-0.3950 & 0.01145\\ -0.011 & 0\end{bmatrix}\begin{bmatrix}H(t)\\ h(t)\end{bmatrix}+\begin{bmatrix}0.03362 & 1.038\\ 0.000966 & 0\end{bmatrix}\begin{bmatrix}u_1(t)\\ u_2(t)\end{bmatrix}\\ \begin{bmatrix}y_1(t)\\ y_2(t)\end{bmatrix}=\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}H(t)\\ h(t)\end{bmatrix}+\begin{bmatrix}0 & 0\end{bmatrix}\begin{bmatrix}u_1(t)\\ u_2(t)\end{bmatrix}\end{cases}$$

试用矩阵组 $[\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}]$ 表示系统。

【解】

由系统的状态空间模型，有以下 MATLAB 语句：

```
a = [-0.3950  0.01145;-0.011 0];
b = [0.03362 1.038;0.000966 0];
c = [1 1];d = [0 0];sys=ss(a,b,c,d)
sys1=tf(sys)
sys2=zpk(sys)
```

执行以上语句可得系统的状态空间模型：

```
a =
                  x1           x2
    x1        -0.395      0.01145
    x2        -0.011            0
b =
                  u1           u2
    x1       0.03362        1.038
    x2      0.000966            0
c =
                  x1           x2
     y1            1            1
d =
                  u1           u2
     y1            0            0
Continuous-time model.
```

执行以上语句还得到系统自第一个输入到输出的传递函数模型为：

```
Transfer function from input 1 to output:
 0.03459 s + 2.281e-005
-------------------------
s^2 + 0.395 s + 0.000126
```

自第二个输入到输出的传递函数模型为：

```
Transfer function from input 2 to output:
   1.038 s - 0.01142
-------------------------
s^2 + 0.395 s + 0.000126
```

自第一个输入到输出的零极点增益模型为：

```
Zero/pole/gain from input 1 to output:
 0.034586 (s+0.0006595)
------------------------
(s+0.3947) (s+0.0003191)
```

自第二个输入到输出的零极点增益模型为：

```
Zero/pole/gain from input 2 to output:
    1.038 (s-0.011)
------------------------
(s+0.3947) (s+0.0003191)
```

【例 4-4】已知二阶离散系统 z 变换传递函数为：

$$\Phi(z)=\frac{1.6z^2-5.8z+3.9}{z^2-0.7z+2.4}$$

试求采样周期 Ts=0.1 s 时系统的传递函数模型与零点极点增益模型。

【解】

根据 MATLAB 程序设计的基本理论，给出以下指令段：

```
num=[1.6 -5.8,3.9]; den=[1,-0.7,2.4];
sys=tf(num,den,0.1)
sys1=zpk(sys)
sys2=filt(num,den,0.1)
```

指令段运行后，即得到系统的传递函数模型、零点极点增益模型与 DSP 形式的传递函数分别为：

```
Transfer function:
1.6 z^2 - 5.8 z + 3.9
---------------------
  z^2 - 0.7 z + 2.4
Sampling time: 0.1
Zero/pole/gain:
1.6 (z-2.733) (z-0.8918)
------------------------
   (z^2 - 0.7z + 2.4)
Sampling time: 0.1
Transfer function:
1.6 - 5.8 z^-1 + 3.9 z^-2
-------------------------
 1 - 0.7 z^-1 + 2.4 z^-2
Sampling time: 0.1
```

【例 4-5】已知离散系统的状态空间方程为：

$$\begin{cases} x(k+1)=\begin{bmatrix}-2.8 & -1.4 & 0 & 0\\ 1.4 & 0 & 0 & 0\\ -1.8 & -0.3 & -1.4 & -0.6\\ 0 & 0 & 0.6 & 0\end{bmatrix}x(k)+\begin{bmatrix}1\\0\\1\\0\end{bmatrix}u(k)\\ y(k)=\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}x(k)\end{cases}$$

试求采样周期 Ts=0.1s 时系统的传递函数模型的两个向量 ***num***、***den*** 与系统的零极点增益模型。

【解】

由系统的状态空间模型，给出以下语句段：

```
a = [-2.8 -1.4 0 0;1.4 0 0 0;-1.8 -0.3 -1.4 -0.6;0 0 0.6 0];
b = [1;0;1;0];c=[0 0 0 1];d=[0];sys=ss(a,b,c,d,0.1);
sys1=tf(sys);num=sys1.num{1}
den=sys1.den{1}
sys2=zpk(sys)
```

运行以上语句段可得采样周期 Ts=0.1s 时系统的传递函数模型的两个向量 ***num***、***den*** 与系统的零极点增益模型分别为：

```
num =
      0          0     0.6000     0.6000     0.9240
den =
   1.0000     4.2000     6.2400     3.7520     0.7056
Zero/pole/gain:
   0.6 (z^2 + z + 1.54)
--------------------------------
(z+1.4)^2 (z+1.061) (z+0.3394)
Sampling time: 0.1
```

【例 4-6】设一高阶系统的传递函数为：

$$G(s)H(s)=\frac{0.0001s^3+0.0218s^2+1.0436s+9.3599}{0.0006s^3+0.0268s^2+0.06365s+6.2711}$$

试将系统的传递函数模型转换为状态空间模型及零极点增益模型。

【解】

这里已知系统的传递函数模型，利用模型转换函数，既可将该系统由传递函数模型转换为状态空间模型，也可以将传递函数模型转换为零极点增益模型。给出以下程序段：

```
num = [0.0001 0.0218 1.0436 9.3599];
den = [0.0006 0.0268 0.6365 6.2711];
sys=tf(num,den);sys1=ss(sys)
sys2=zpk(sys)
```

程序段运行后，即得到系统状态空间模型的状态矩阵 ***a***；系统控制矩阵 ***b***；系统输出

矩阵 $\boldsymbol{c}$；系统直接传输矩阵 $\boldsymbol{d}$：

```
a =
                    x1          x2          x3
          x1   -44.667     -16.576     -10.207
          x2        64           0           0
          x3         0          16           0
b =
                    u1
          x1         8
          x2         0
          x3         0
c =
                    x1          x2          x3
          y1    3.6111      3.0518      1.6916
d =
                    u1
          y1   0.16667
Continuous-time model.
```

由以上数据可写出系统的状态空间模型：

$$\begin{cases} \dot{x}(t)=\begin{bmatrix} -44.667 & -16.576 & -10.207 \\ 64 & 0 & 0 \\ 0 & 16 & 0 \end{bmatrix}\boldsymbol{x}(t)+\begin{bmatrix} 8 \\ 0 \\ 0 \end{bmatrix}\boldsymbol{u}(t) \\ \boldsymbol{y}(t)=[3.6111 \quad 3.0518 \quad 1.6916]\boldsymbol{x}(t)+0.1667\boldsymbol{u}(t) \end{cases}$$

程序段运行后还得到系统的零点极点增益模型：

```
Zero/pole/gain:
0.16667 (s+154.3) (s+52.05) (s+11.65)
-----------------------------------
  (s+17.99) (s^2 + 26.67s + 580.9)
```

4.1.2.4 第四种系统数学模型

不一定非要有用数学表达式描述的数学模型，才能对系统进行仿真。恰恰相反，就有不用数学表达式的数学模型，能对系统进行仿真的。在 MATLAB 中特有另外一种数学模型即 SIMULINK 模型窗口（叫做“Untitled” 窗口）里的动态结构图。只要在 SIMULINK 工作窗里，按其规则（参见第 3 章）画出动态结构图，就是对系统建立了数学模型。再按规则将结构图的参量用实际系统的数据进行设置，就可直接对系统进行仿真了。而且用 SIMULINK 模型窗口里的动态结构图可直接、方便地对系统进行各种仿真，这种方法使用极其广泛而普遍。

这种数学模型在 3.3 节中已经作了详尽介绍。

4.1.3 环节方框图模型的化简

系统是受控对象与控制装置组成的，即系统由多个环节组成。每个环节又是由多个元件构成的。环节在 MATLAB 里又叫模块。自动控制的对象可以是一个元件，一个环节，也可以是一个模块，一个装置，甚至是一个系统。这要根据讨论问题的实际情况来确定。系统方框图模型的化简也适用于环节的、模块的、装置的方框图模型的化简。以下将以环节的各种连接来加以叙述。

4.1.3.1 环节串联连接的化简

多个环节相串联的连接形式是控制系统最基本的组成结构形式之一。环节串联是指前一个环节的输出为相邻的下一个环节的输入，依此类推（图 4-3）。这种结构形式与多个电阻串联相类似。所谓环节串联的化简就是求以第一个环节的输入为等效传递函数的输入，以最后一个环节的输出为其等效输出的等效结构。在自动控制中，环节串联的等效传递函数为各个串联环节的传递函数的乘积。这与多个电阻串联其等效电阻为其多个电阻之和是不能混为一谈的。将串联的多个环节的传递函数方框，在 SIMULINK 的模型窗口里依次串接画出即成为系统方框图模型。控制系统的环节串联及其化简就是模块方框图模型的串联及其化简。可以用 MATLAB 的函数命令 series ( ) 将串联模块化简。

$u_1(t)$ → $G_1(s)$ → $y_1(t)$ $u_2(t)$ → $G_2(s)$ → $y_2(t)$

图 4-3 串联连接结构

series ( ) 函数命令可以将两个环节串联连接进行等效化简。它既适用于连续时间系统，也适用于离散时间系统。

如果已知两个环节的传递函数分别为：$G_1=\dfrac{\boldsymbol{num}1(s)}{\boldsymbol{den}1(s)}$，$G_2=\dfrac{\boldsymbol{num}2(s)}{\boldsymbol{den}2(s)}$，则两个环节串联连接的等效传递函数为：

$$G(s)=\frac{\boldsymbol{num}(s)}{\boldsymbol{den}(s)}=G_1(s)G_2(s)=\frac{\boldsymbol{num}1(s)}{\boldsymbol{den}1(s)}\frac{\boldsymbol{num}2(s)}{\boldsymbol{den}2(s)} \tag{4-11}$$

使用 series ( )函数命令，不必做多项式的乘除运算即可实现两个环节传递函数（sys1 与 sys2）串联连接即实现式(4-11)的运算。如果 sys1=tf (***num***1,***den***1), sys2=tf (***num***2,***den***2)，其命令格式为：

sys=series(sys1,sys2)

如果已知两个环节状态空间模型的矩阵组分别为：（*a*1,*b*1,*c*1,*d*1）与（*a*2,*b*2,*c*2,*d*2），则求两个环节串联连接等效系统状态空间模型[***a,b,c,d***]矩阵组的命令格式为：

[*a*,*b*,*c*,*d*]=series(*a*1,*b*1,*c*1,*d*1,*a*2,*b*2,*c*2,*d*2)

此时，是将环节 1 的所有输出连接到环节 2 的输入上，即 $\boldsymbol{y}_1=\boldsymbol{u}_2$。串联环节的等效系统状态空间表达式为：

$$\begin{cases}\begin{bmatrix}\dot{x}_1\\\dot{x}_2\end{bmatrix}=\begin{bmatrix}a_1 & 0\\b_2c_1 & a_2\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}b_1\\b_2d_1\end{bmatrix}u_1\\ y_2=\begin{bmatrix}d_2c_1 & c_2\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}d_2d_1\end{bmatrix}u_1\end{cases} \tag{4-12}$$

需要特别指出，series ( ) 函数命令还可以将多个环节按两两串联的形式多次递归调用加以连接，进行等效化简。

实际上 sys=series(sys1,sys2) 命令现在很少用，已由命令 **sys=sys1*sys2*…*sysn** 所取代，这样命令中不仅省掉了“series ( )”字符，而且可以实现多个传递函数模块的串联，一次就可以求出其等效传递函数。

【例 4-7】已知双环调速系统电流环内前向通道三个模块的传递函数分别为：

$$G_1(s)=\frac{0.0128\,s+1}{0.04\,s}，\quad G_2(s)=\frac{30}{0.00167s+1}，\quad G_3(s)=\frac{2.5}{0.0128\,s+1}$$

试求串联连接的等效传递函数及等效空间模型。

【解】

根据 MATLAB 程序设计的基本理论与系统串联连接的函数命令 series ( ) ,可以编写出 MATLAB 程序 L4107.m 如下。

```
% MATLAB PROGRAM L4107.m
%
n1=[0.0128 1];d1=[0.04 0];sys1=tf(n1,d1);n2=[30];d2=[0.00167 1];
sys2=tf(n2,d2);  n3=[2.5];d3=[0.0128 1];sys3=tf(n3,d3);
s1=ss(sys1);s2=ss(sys2);s3=ss(sys3);
sys123=sys1*sys2*sys3
sys12=series(s1,s2); sys=series(sys12,s3)
```

程序运行后有电流环内前向通道的等效传递函数及等效状态空间模型：

```
Transfer function:
            0.96 s + 75
-------------------------------------
8.55e-007 s^3 + 0.0005788 s^2 + 0.04 s
a =
                    x1          x2          x3
          x1   -78.125      2245.5           0
          x2         0      -598.8         800
          x3         0           0           0
b =
                    u1
          x1         0
          x2     40.96
          x3         4
```

```
c =
                    x1          x2          x3
        y1      12.207           0           0
d =
                    u1
        y1           0
Continuous-time model.
```

由以上运算数据可以写出系统等效传递函数为：

$$G(s)=G_1(s)G_2(s)G_3(s)=\frac{0.96s+75}{0.000000855\,s^3+0.0005788\,s^2+0.04s}$$

系统等效空间模型为：

$$\begin{cases}\dot{x}(t)=\begin{bmatrix}-78.125 & 2245.5 & 0\\ 0 & -598.8 & 800\\ 0 & 0 & 0\end{bmatrix}x(t)+\begin{bmatrix}0\\ 40.96\\ 4\end{bmatrix}u(t)\\ y(t)=\begin{bmatrix}12.207 & 0 & 0\end{bmatrix}x(t)\end{cases}$$

4.1.3.2 环节并联连结的化简

多个环节相并联的连接形式也是控制系统最基本的组成结构形式之一。环节并联是指多个环节的输入信号相同，所有环节输出的代数和为其总输出。两环节或模块相并联见图 4-4。

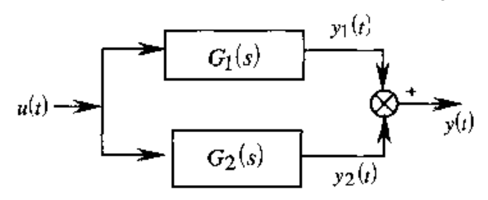


图 4-4　模块的并联连接

parallel（）函数命令可以将两个环节并联连接进行等效化简。它既适用于连续时间系统也适用于离散时间系统。parallel（）函数命令调用格式有：

[*num*, *den*] = parallel(*num*1,*den*1,*num*2,*den*2)

求出的等效传递函数为：

$$G(s)=\frac{num(s)}{den(s)}=G_1(s)+G_2(s)=\frac{num1(s)den2(s)+num2(s)den1(s)}{den1(s)den2(s)} \tag{4-13}$$

实际上这个命令现在很少用，已由命令 sys=sys1+sys2+...+sysn 所取代，这样命令中不仅省掉了“parallel( )”字符，而且可以实现多个传递函数模块的并联求其代数和，一次就可以求出其等效传递函数。parallel（）函数命令调用格式还有：

[a,b,c,d] = parallel(a1,b1,c1,d1,a2,b2,c2,d2)

命令执行后可以得到由环节 1 与环节 2 并联连接的状态空间模型表示的系统，该系统为：

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \boldsymbol{u} \\ y_2 = y_1 + y_2 = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} d_1 & d_2 \end{bmatrix} \boldsymbol{u} \end{cases} \tag{4-14}$$

这两种命令调用格式都是将两环节的输入连接在一起并作为系统的输入，其输出为 $y = y_1 + y_2$，如图 4-4 所示。还需要指出，parallel ( ) 函数命令也可以将多个环节按两两并联递归调用加以连接，进行等效化简。

【例 4-8】已知两子系统传递函数分别为：

$$G_1(s) = \frac{5}{s+1}, \quad G_2(s) = \frac{7s+8}{s^2+2s+9}$$

试求两系统并联连接的等效传递函数的 ***num*** 与 ***den*** 向量及等效空间模型。

【解】

根据 MATLAB 程序设计的基本理论与系统并联连接化简的函数命令 parallel ( )，给出 MATLAB 程序 L4108.m 如下。

```
%MATLAB PROGRAM L4108.m
%
num1=[5];den1=[1,1];sys1=tf(num1,den1);
num2=[7,8];den2=[1,2,9];sys2=tf(num2,den2);
s1=ss(sys1);s2=ss(sys2);
sys=sys1+sys2;num=sys.num{1}
den=sys.den{1}
sysz=parallel(s1,s2)
```

运行程序后有等效传递函数的 ***num*** 与 ***den*** 向量及等效空间模型为：

```
num =
    0    12    25    53
den =
    1     3    11     9
a =
           x1     x2     x3
    x1     -1      0      0
    x2      0     -2  -2.25
    x3      0      4      0
b =
           u1
    x1      2
    x2      4
```

```
    x3      0
c =
          x1     x2     x3
    y1   2.5  1.75   0.5
d =
          u1
   y1     0
Continuous-time model.
```

由以上运算数据可以写出系统等效传递函数为：

$$G(s)=G_1(s)+G_2(s)=\frac{12s^2+25s+53}{s^3+3s^2+11s+9}$$

系统等效空间模型为：

$$\begin{cases}\dot{x}(t)=\begin{bmatrix}-1 & 0 & 0\\ 0 & -2 & -2.25\\ 0 & 4 & 0\end{bmatrix}x(t)+\begin{bmatrix}2\\4\\0\end{bmatrix}u(t)\\ y(t)=[2.5 \quad 1.75 \quad 0.5]x(t)\end{cases}$$

4.1.3.3　环节反馈连结的化简

两个环节的反馈连接见图 4-5。控制系统环节的反馈连接及其等效化简就是模块方框图模型的反馈连接及其化简。

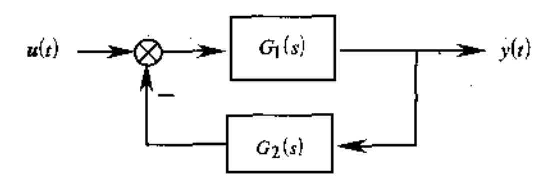


图 4-5　反馈连接结构

若 $G_1(s)=\dfrac{\boldsymbol{num}1(s)}{\boldsymbol{den}1(s)}$，　$G_2(s)=\dfrac{\boldsymbol{num}2(s)}{\boldsymbol{den}2(s)}$，反馈连接结构的等效传递函数为：

$$\Phi(s)=\frac{\boldsymbol{num}(s)}{\boldsymbol{den}(s)}=\frac{G_1(s)}{1\mp G_1(s)G_2(s)}=\frac{\boldsymbol{num}1(s)\boldsymbol{den}2(s)}{\boldsymbol{den}1(s)\boldsymbol{den}2(s)\mp \boldsymbol{num}1(s)\boldsymbol{num}2(s)} \quad (4\text{-}15)$$

若两个环节都用状态空间模型表示，反馈连接后产生的系统为：

$$\begin{cases}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}a_1+b_1ed_2c_1 & \pm b_1ec_2\\ b_2c_1\pm b_2d_1ed_2c_1 & a_2\pm b_2d_1ec_1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}b_1(I\pm ed_2d_1)\\ b_2d_1(I\pm ed_2d_1)\end{bmatrix}u_1\\ y_2=y_1+y_2=[c_1\pm d_1ed_2c_1 \quad \pm d_1ec_2]\begin{bmatrix}x_1\\ x_2\end{bmatrix}+[d_1(I\pm ed_2d_1) \quad d_2]u_1\end{cases} \quad (4\text{-}16)$$

式中

$$e = \left(I \mp d_2 d_1\right)^{-1} \tag{4-17}$$

MATLAB 中的 feedback（）函数命令可将两个环节按反馈形式进行连接后求其等效传递函数。G1 为闭环前向通道的传递函数，G2 为反馈通道的传递函数。feedback ( ) 函数既适用于连续时间系统，也适用于离散时间系统。

feedback ( ) 函数命令格式为：

G=**feedback**(*G*1,*G*2,*sign*)

[*a*,*b*,*c*,*d*] = **feedback**(*a*1,*b*1,*c*1,*d*1,*a*2,*b*2,*c*2,*d*2,*sign*)

[***num***, ***den***] = **feedback**(***num***1,***den***1,***num***2,***den***2,*sign*)

G=feedback(*G*1,*G*2,*sign*)函数可将两个环节按反馈方式连接起来，环节 *G*1 的所有输出均连接到环节 2 的输入，环节 *G*2 的所有输出均连接到环节 *G*1 的输入，*sign* 是指 y_2 到 u_1 连接的符号，*sign* 缺省时，默认为负，即 *sign*=-1。

[***num***, ***den***]=feedback(***num***1,***den***1,***num***2,***den***2,*sign*) 可得到类似的连接，只是子系统与闭环传递函数都是用分子分母 s 多项式的形式表示的。现在这个命令格式很少使用了，已被 *G*=feedback(*G*1,*G*2,*sign*)格式所取代。

[*a*,*b*,*c*,*d*] = feedback(*a*1,*b*1,*c*1,*d*1,a2,*b*2,*c*2,*d*2,*sign*)也可得到类似的连接，只是子系统与闭环等效系统都是用状态空间模型矩阵组的形式表示的。

【例 4-9】已知晶闸管-直流电机单闭环调速系统（V-M 系统）的 SIMULINK 动态结构图如图 4-6 所示。

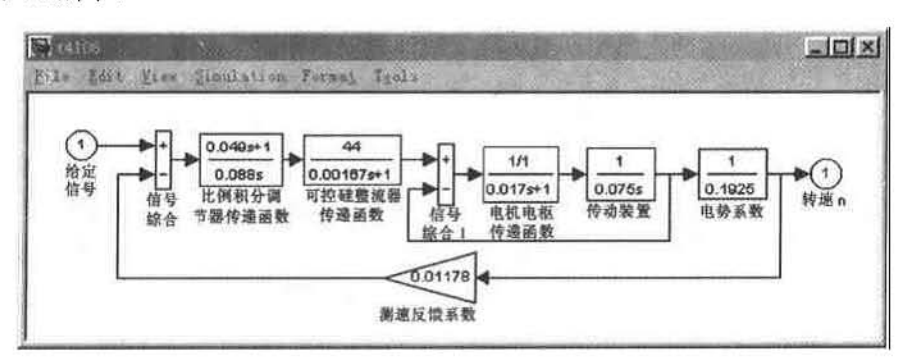


图 4-6 直流单闭环调速系统

求按负反馈连接闭环系统的传递函数。

【解】

根据 MATLAB 程序设计的基本理论给出程序 L4109.m 如下。

```
% MATLAB PROGRAM L4109.m
%
n1=[1];d1=[0.017 1];s1=tf(n1,d1);
n2=[1];d2=[0.075 0];s2=tf(n2,d2);
sys1=feedback(s1*s2,1)
n3=[0.049 1];d3=[0.088 0];s3=tf(n3,d3);
```

```
n4=[C 44];d4=[0.00167 1];s4=tf(n4,d4);
n5=1;d5=0.1925;s5=tf(n5,d5);
n6=0.01178;d6=1;s6=tf(n6,d6);
sysq=sys1*s3*s4*s5;
sys=feedback(sysq,s6)
```

程序运行后有：

```
Transfer function:
              1
--------------------------
0.001275 s^2 + 0.075 s + 1
```

由以上运算数据可得单闭环系统内的小闭环的传递函数为：

$$\Phi_1(s)=\frac{1}{0.001275s^2+0.075s+1}$$

单闭环系统的闭环传递函数（当略去分母的 s^4 及以上项后）为：

$$\Phi(s)\approx\frac{0.0022749s^2+0.2178s+3.33}{0.000001218s^3+0.0001052s^2+0.00444s+0.05581}$$

4.1.4 SIMULINK 动态结构图举例

前面已经介绍了三种数学表达式系统数学模型及其互相转换。作为系统第四种数学模型的 SIMULINK 结构图，通常只与传递函数模型关系密切。因为 SIMULINK 动态结构图里画的模块就是环节的传递函数。画系统的 SIMULINK 动态结构图的方法也只是在 SIMULINK 窗口里拷贝系统的各个标准模块（环节）的方框然后将其连接成系统。

在此举一例简介如下。图 4-7 是直流双闭环调速系统 SIMULINK 动态结构图，为直流电机传动、电流负反馈内环接转速负反馈外环的双环调速系统。系统单输入，输入量为转速给定电压信号；系统单输出，输出量为电机转速。系统里有电流调节器与转速调节器，以这两个调节器构成电流负反馈与转速负反馈的闭环控制。

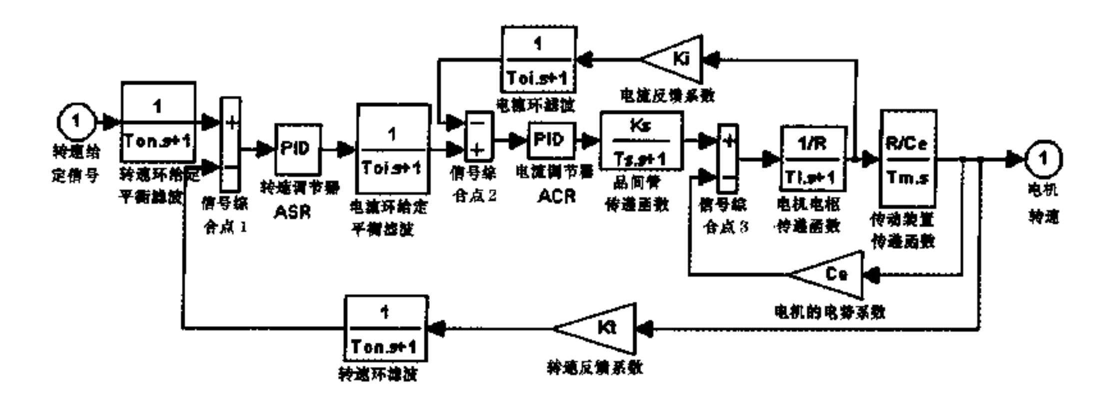


图 4-7　双闭环调速系统 SIMULINK 动态结构图

有了系统的动态结构图数学模型，就可以对系统进行各种不同的仿真。正如在 3.5 节已介绍的，既可以用菜单操作方式，也可以用在 MATLAB 命令窗口中的指令工作方式。

4.2 控制系统稳定性分析的 MATLAB 实现

4.2.1 控制系统稳定性的有关概念

为了讨论控制系统稳定性，有必要简单复习有关系统稳定性的几个基本概念。

1．控制系统的特征多项式与特征方程

设线性定常系统闭环传递函数为：

$$\Phi(s)=\frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_{m-1}s + b_m}{a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n}=\frac{M(s)}{D(s)} \tag{4-18}$$

式中

$$D(s)=a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n \tag{4-19}$$

式（4-19）叫做系统特征多项式。令 $D(s)=0$，此为系统特征方程。

2．系统稳定性判别的特征方程法

系统稳定的必要充分条件是系统特征方程的全部根，或者系统闭环传递函数的全部极点都位于 s 左半平面。

3．Hurwitz 稳定判据

设线性定常系统特征方程为：

$$D(s)=a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n = 0 \quad (a_0>0) \tag{4-20}$$

则系统稳定的必要充分条件是：特征方程的各项系数均大于 0，且如下古尔维茨行列式全部大于 0：

$$\boldsymbol{D}_1 = a_1$$

$$\boldsymbol{D}_2 = \begin{vmatrix} a_1 & a_3 \\ a_0 & a_2 \end{vmatrix}$$

$$\boldsymbol{D}_3 = \begin{vmatrix} a_1 & a_3 & a_5 \\ a_0 & a_2 & a_4 \\ 0 & a_1 & a_3 \end{vmatrix}$$

……

$$\boldsymbol{D}_n = \begin{vmatrix} a_1 & a_3 & a_5 & \cdots & a_{2n-1} \\ a_0 & a_2 & a_4 & \cdots & a_{2n-2} \\ 0 & a_1 & a_3 & \cdots & a_{2n-3} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & a_n \end{vmatrix}$$

4．结构不稳定系统

不改变系统结构仅靠调整其各元部件的参数而无法保证稳定的系统。

4.2.2 控制系统稳定性的基本概念与分析方法简述

Lyapunov为研究运动的稳定性提出了两种方法，第一种方法属于经典控制的范畴，就是通过求解微分方程组的显式解来对系统稳定性进行分析；第二种方法是不求解微分方程组，而通过构造李雅普诺夫函数来判断系统运动的稳定性。这种方法属于状态空间分析的范畴。

经典控制分析中，关于线性定常系统稳定性的概念是，如果系统受到扰动的作用而偏离了原平衡状态。当扰动去除后，若能够恢复到原来平衡状态，系统就是稳定的；否则系统就是不稳定的。线性定常系统稳定性的数学定义是控制系统闭环特征方程的全部根，不论是实根或复根，其实部均应为负值，闭环系统就是稳定的。由此可知，求解控制系统闭环特征方程的根并进而判断所有根的实部是否小于零，这种方法成为控制系统稳定性判别最基本的方法。

基于判别特征方程根的方法，还有很多其他与此相关的系统稳定性判别方法，都叫做代数稳定判据。有基于求系统特征方程古尔维茨行列式的 **Hurwitz** 稳定判据，有列劳斯表的 Routh 稳定判据等等。还有根据系统开环传递函数绘制闭环特征根轨迹的方法，直接从s平面上根轨迹就可以判别闭环系统是否稳定。

在频域分析里，根据系统开环 Nyquist 曲线$G(j\omega)$相对于(-1，j0)点的位置关系，有判别闭环系统稳定性的 Nyquist 稳定判据。还有根据系统开环 Bode 图幅频与相频曲线相互位置关系的对数频率稳定判据。

在状态空间分析里，有用李雅普诺夫函数概念阐述稳定性的 **Lyapunov** 稳定定理。

有如此之多的判据，从各个不同的角度对系统的稳定性加以判别，说明系统稳定性是系统能够成立与运行的首要条件。设计或创建的系统，首先必须是稳定的。只有稳定的系统，才能讨论与研究系统自动控制的其他问题。例如只有判稳过的系统，才计算稳态误差。所以控制系统的稳定性分析是系统时域分析、频率分析、根轨迹分析、稳态误差分析的前提，对于设计或创建的系统，首要的问题就是进行稳定性分析。

4.2.3 稳定性分析 MATLAB 实现的方法与举例

控制系统稳定性分析的各种方法，各有其特长与应用范围。以下就代数稳定判据、根轨迹法、Bode 图法判断系统稳定性的 MATLAB 实现进行介绍。

4.2.3.1 代数稳定判据与举例

求解控制系统闭环特征方程的根并判断所有根的实部是否小于零，在 MATLAB 里这是很容易用函数 roots ( ) 实现的。求多项式根的函数 roots ( ) 在 2.3 节里已经介绍过。

roots (***P***) 函数输入参量 ***P*** 是降幂排列多项式系数向量，在自动控制的稳定性分析中，***P*** 就是系统闭环特征多项式降幂排列的系数向量。若能够求得 ***P***，则其根就可以求出。

【例 4-10】已知系统的开环传递函数为：

$$G(s)=100\frac{(s+2)}{s(s+1)(s+20)}$$

试对系统闭环判别其稳定性。

【解】

根据题意，利用 roots ( ) 函数给出以下 MATLAB 程序段：

```
k=100;z=[-2];p=[0,-1,-20];
[n1,d1]=zp2tf(z,p,k);
G=tf(n1,d1);
P=n1+d1
roots(P)
```

运行该程序段可得多项式系数向量 ***P*** 及其根：

```
P =
     1    21    120    200
ans =
  -12.8990
   -5.0000
   -3.1010
```

计算数据表明所有特征根的实部均为负值，所以闭环系统是稳定的。

【例 4-11】已知系统的动态结构图模型如图 4-8 所示，试对系统闭环判别其稳定性。

【解】

根据题意，调用 roots ( ) 函数给出以下 MATLAB 程序段：

```
n1=[0 0 10];d1=[1 1 0];s1=tf(n1,d1);
n2=[0 2 0];d2=[0 0 1];s2=tf(n2,d2);
s12=feedback(s1,s2);
n3=[0 1 1];d3=[0 1 0];s3=tf(n3,d3);
sys1=s12*s3;
sys=feedback(sys1,1);
roots(sys.den{1})
```

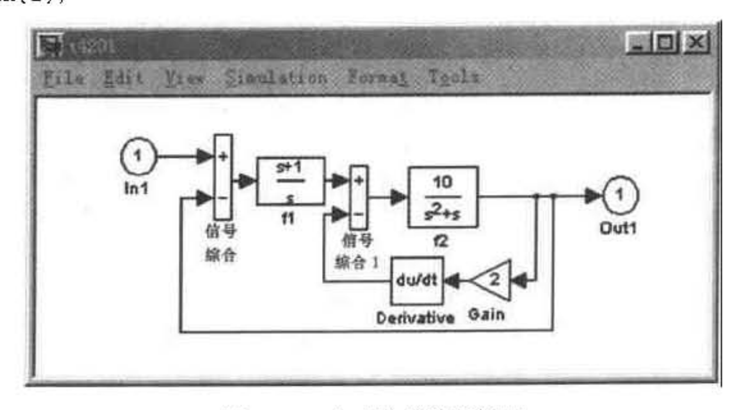


图 4-8　已知系统结构图模型

运行该程序段可得闭环特征多项式系数向量 ***P*** 的根：

```
ans =
-20.5368
 -0.2316 + 0.6582i
 -0.2316 - 0.6582i
```

计算数据表明所有特征根的实部均为负值，所以闭环系统是稳定的。

【例 4-12】已知系统的闭环传递函数为：

$$\Phi(s)=\frac{5s+200}{0.001s^3+0.502s^2+6s+200}$$

试对系统闭环判别其稳定性。

【解】

根据题意，利用 roots ( ) 函数给出以下 MATLAB 程序段：

```
n1=[5 200];d1=[0.001 0.502 6 200];s1=tf(n1,d1);
G=zpk(s1);G.p{1}
```

运行该程序段可得闭环特征多项式系数向量 ***P*** 的根：

```
ans =
  1.0e+002 *
  -4.9060
  -0.0570 + 0.1937i
  -0.0570 - 0.1937i
```

计算数据表明所有特征根的实部均为负值，所以闭环系统是稳定的。运行以下另一段程序可得到同样的结果。

```
n1=[5 200];d1=[0.001 0.502 6 200];s1=tf(n1,d1);
G=ss(s1);eig(G.a)
```

4.2.3.2　用根轨迹法判断系统稳定性及其举例

在 4.6 节中将要介绍的函数 rlocfind（）有一种格式的功能可以求出系统闭环的所有极点，即函数格式[*k,poles*] = rlocfind (*sys*)。输入参量 *sys* 可以是 LTI 模型对象，函数命令执行后，可在图形窗口根轨迹图中显示十字形光标，当选择根轨迹上某一点时，其相应的增益由变量 *k* 记录，与此增益相关的所有极点记录在变量 *poles* 中。从显示所有极点的数值（即位置），就可判断系统的稳定性。

【例 4-13】已知一个单位负反馈系统开环传递函数为 $G(s)$，试在系统闭环的根轨迹图上选择一点，求出该点的增益 *k* 及其系统的闭环极点位置，并判断在该点系统闭环的稳定性。

$$G(s)=k\frac{(s+3)}{s(s+5)(s+6)(s^2+2s+2)}$$

【解】

根据题目要求，调用函数命令 rlocfind ( ) 的程序 L4204.m 如下。

```
% MATLAB PROGRAM L4204.m
%
num=[1 3];den=conv(conv(conv([1 0],[1 5]),[1 6]),[1 2 2]);
sys=tf(num,den);rlocus(sys)
[k,poles]=rlocfind(sys)
```

该程序执行后可得系统闭环的根轨迹，如图 4-9 所示。

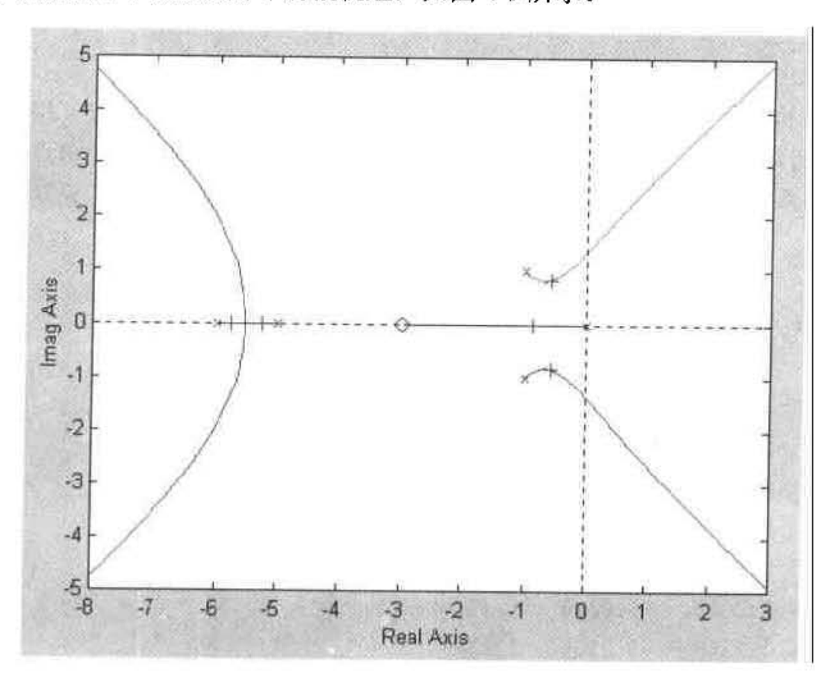


图 4-9　系统闭环的根轨迹图

该程序执行后还在根轨迹图窗口上显示十字形光标，当在根轨迹负实轴上选择一点时，其相应的增益由变量 k 记录，与增益相关的所有极点记录在变量 *poles* 中。其数据是：

```
k =
   8.6955
poles =
  -5.7727
  -5.2653
  -0.5525 + 0.8344i
  -0.5525 - 0.8344i
  -0.8570
```

由计算数据可知，系统全部闭环极点的实部都为负值，所以在该点(k =8.6955)闭环系统是稳定的。

【例 4-14】续【例 4-13】，试计算当 k 在 33~37 范围内时系统的闭环极点位置，并判断系统闭环的稳定性。

【解】

根据题目要求，用函数命令编写 MATLAB 程序 L4205a.m。

```
%MATLAB PROGRAM L4205a.m
%
num=[1 3];den=conv(conv(conv([1 0],[1 5]),[1 6]),[1 2 2]);
cpole=rlocus(num,den,[33:1:37]);
range=[33:1:37]';
[range,cpole]
```

该程序执行后可得单位反馈系统 k 在 33~37 范围内闭环极点位置（根轨迹上）的数据：由闭环极点位置的数据可以看出，当 $k<35$ 时，闭环共轭复数极点位于复平面虚轴以左，系统处于稳定状态；当 $k>36$ 时，闭环特征方程出现右根，系统处于不稳定状态。

```
ans =
  Columns 1 through 4
  33.0000  -5.5745 + 0.6697i  -5.5745 - 0.6697i  -1.7990
  34.0000  -5.5768 + 0.6850i  -5.5768 - 0.6850i  -1.8154
  35.0000  -5.5791 + 0.7001i  -5.5791 - 0.7001i  -1.8313
  36.0000  -5.5815 + 0.7147i  -5.5815 - 0.7147i  -1.8466
  37.0000  -5.5838 + 0.7291i  -5.5838 - 0.7291i  -1.8615
  Columns 5 through 6
  -0.0260 + 1.3210i  -0.0260 - 1.3210i
  -0.0155 + 1.3340i  -0.0155 - 1.3340i
  -0.0052 + 1.3467i  -0.0052 - 1.3467i
   0.0048 + 1.3591i   0.0048 - 1.3591i
   0.0146 + 1.3712i   0.0146 - 1.3712i
```

可以用以下程序(L4205b.m)来校核当 k=35 与 k=36 时闭环系统的阶跃给定响应曲线：

```
% MATLAB PROGRAM L4205b.m
%
n1=[1 3];d1=conv(conv(conv([1 0],[1 5]),[1 6]),[1 2 2]);
for k=35:36
n=k*n1;
    s1=tf(n,d1);
    G=feedback(s1,1);
    step(G);hold on
end
gtext('k=35'),gtext('k=36')
```

程序运行后得到系统闭环单位阶跃给定响应曲线，如图 4-10 所示。当 k=35 时，曲线呈强烈的衰减振荡，系统闭环还算稳定；当 k=36 时，曲线振荡发散，系统闭环不稳定。

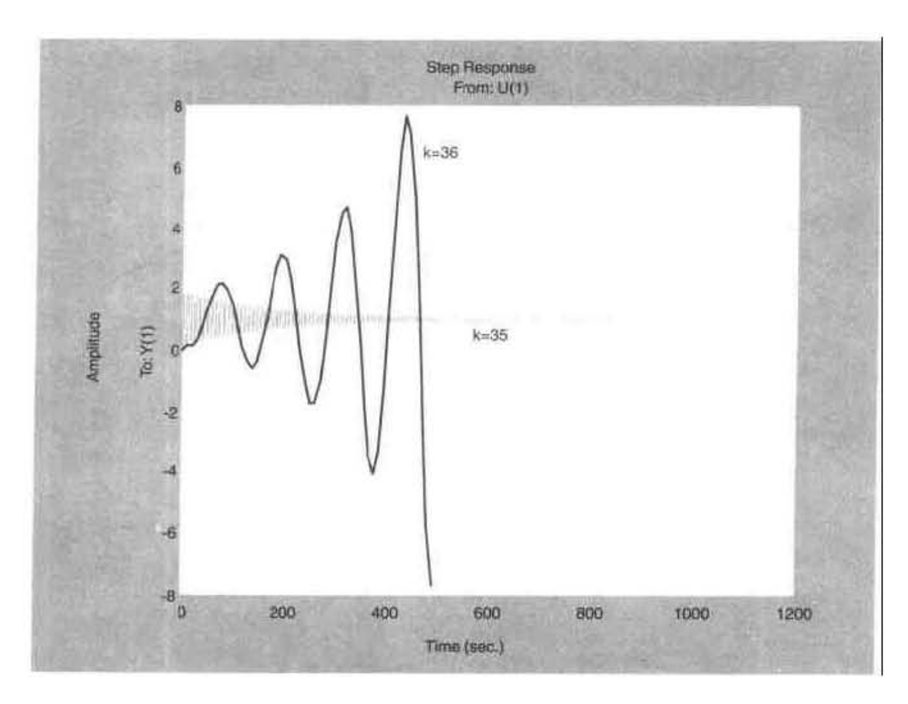


图 4-10 闭环系统单位阶跃给定响应曲线

4.2.3.3 用 Bode 图法判断系统稳定性及其举例

有了系统的 Bode 图，就可以计算频域性能指标。在 4.5 节中将要介绍的求系统模值裕度与相位裕度的函数 margin ()，既可绘制系统 Bode 图，又能够计算频域性能指标。

【例 4-15】已知两个单位负反馈系统开环传递函数分别为：

$$G_1(s)=\frac{2.7}{s^3+5s^2+4s}\ ;\quad G_2(s)=\frac{2.7}{s^3+5s^2-4s}$$

用 Bode 图法判断系统闭环的稳定性。

【解】

根据题目要求，对第一个系统用 margin ( ) 函数命令给出如下程序 L4206a.m。

```
%MATLAB PROGRAM L4206a.m
%
num=[0 0 0 2.7];den=[1 5 4 0];s1=tf(num,den);
[Gm,Pm,Wcp,Wcg]=margin(s1)
margin(s1)
```

该程序运行后，得第一个系统的 Bode 图（图 4-11）与计算的频域性能指标：

模值稳定裕度: G_m = 7.4074 即 20*log10(7.4074)=17.3933 dB

相位稳定裕度: P_m = 51.7320°

-π 穿越频率: ω_{cp} = 2 1/s

剪切频率: ω_{cg} = 0.5783 1/s

这些频域性能指标数据说明系统闭环不仅稳定，而且有很大的稳定裕度。

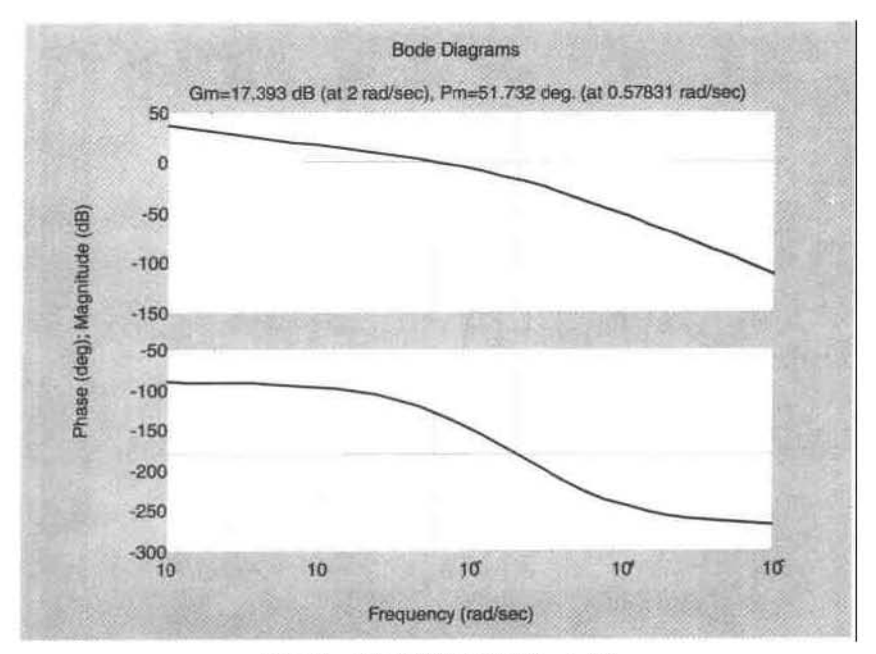


图 4-11 稳定闭环系统的 Bode 图

同样对第二个系统给出 L4206b.m 如下。

```
% MATLAB PROGRAM L4206b.m
%
n2=[0 0 0 2.7];d2=[1 5 -4 0];s2=tf(n2,d2);
margin(s2)
[Gm,Pm,Wcp,Wcg]=margin(s2)
```

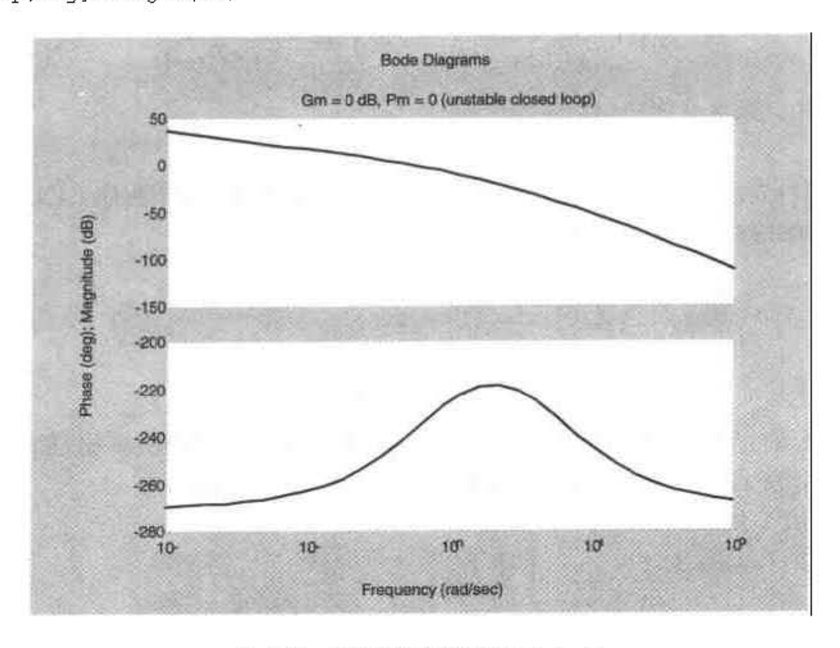


图 4-12 不稳定闭环系统的 Bode 图

程序运行后可得到系统的 Bode 图（图 4-12）与计算的频域性能指标：
模值稳定裕度：G_m = 1 即 20*log10(1)=0 dB
相位稳定裕度：P_m = 0°

频域性能指标数据说明系统闭环是不稳定的。在图 4-12 上，MATLAB 已经给出了系统闭环是“unstable closed loop”即不稳定的结论。

4.3 控制系统稳态误差分析的 MATLAB 实现

4.3.1 控制系统稳态误差分析的有关概念

为了讨论控制系统稳定性，在此简要复习有关系统稳定性的几个基本概念。

1. 稳态误差 e_{ss}

定义为稳定系统误差的终了值：

$$e_{ss}=\lim_{t\to\infty}e(t) \tag{4-21}$$

对于阶跃响应，指阶跃响应曲线 $h(t)$ 的稳态值与期望值之差。即

$$e_{ss}=1-h(\infty) \tag{4-22}$$

它是系统的稳态时域性能指标。需要特别指出，在计算系统稳态误差之前，必须对系统判稳。只有稳定的系统，计算稳态误差才有意义。

2. 自动控制系统的型别

闭环控制系统的开环传递函数里积分环节的个数叫做控制系统的型别，或者称为控制系统的无差度。系统的开环传递函数里有一个积分环节的叫 I 型系统；有两个积分环节的叫 II 型系统；其余依次类推。

3. 控制系统的稳态误差系数

稳态误差系数是表明系统在典型外作用下稳态精度的指标，误差系数越大，误差越小，精度越高。常用的有三种误差系数。

（1）稳态位置误差系数 K_p

$$K_p=\lim_{s\to 0}G(s)H(s) \tag{4-23}$$

K_p 表明系统在阶跃给定输入下的稳态精度。

（2）稳态速度误差系数 K_v

$$K_v=\lim_{s\to 0}s\,G(s)H(s) \tag{4-24}$$

K_v 表明系统在斜坡给定输入下的稳态精度。

（3）稳态加速度误差系数 K_a

$$K_a=\lim_{s\to 0}s^2\,G(s)H(s) \tag{4-25}$$

K_a 表明系统在等加速给定输入下的稳态精度。

以上三式中，G(s)H(s)是系统的开环传递函数。

4.3.2 控制系统稳态误差计算

稳态误差计算的原理是拉普拉斯变换的终值定理。其计算公式为：

$$e_{ss}=\lim_{t\to\infty}e(t)=\lim_{s\to 0}sE(s) \tag{4-26}$$

请务必注意拉普拉斯变换终值定理的应用条件：误差 $e(t)$ 的拉氏变换 $E(s)$ 在 s 右平面及除原点外的虚轴上处处解析，即没有极点。

4.3.2.1 三种典型外作用信号输入响应与稳态误差计算

控制系统稳态误差的实际计算，是基于响应曲线的稳态值与期望值之差。对于典型外作用中的单位阶跃信号，MATLAB 中提供了 step ( ) 函数。其稳态误差为阶跃响应曲线 $h(t)$ 的稳态值与期望值（1）之差，即式(4-22)。

另外，几个最常用的信号例如单位斜坡信号与等加速信号作用下的系统稳态误差也可以按这个思路进行计算。

MATLAB 里没有求斜坡响应的函数，为了计算其稳态误差，仍考虑用 step ( ) 函数求系统单位斜坡响应。根据系统闭环传递函数的定义有：

$$\Phi(s)=\frac{C(s)}{R(s)} \tag{4-27}$$

对于单位阶跃信号有：

$$R(s)=\frac{1}{s} \tag{4-28}$$

$$C(s)=\Phi(s)\cdot R(s)=\Phi(s)\cdot\frac{1}{s} \tag{4-29}$$

对于单位斜坡信号有：

$$R(s)=\frac{1}{s^2} \tag{4-30}$$

$$C(s)=\Phi(s)\cdot R(s)=\Phi(s)\cdot\frac{1}{s^2}=\left[\Phi(s)\cdot\frac{1}{s}\right]\cdot\frac{1}{s} \tag{4-31}$$

由式(4-31）看出，可以将系统闭环传递函数除以拉氏算子“s”，再使用 step ( ) 函数计算的就不是阶跃响应，求出的则是系统单位斜坡响应。在 MATLAB 程序中，这只要在系统闭环传递函数的分母多项式乘以“s”即可。

为了计算系统的稳态误差，还需在绘制系统单位斜坡输入信号的响应曲线的同时，绘制出 $r(t)=v_0t\big|_{v_0=1}=t$ 单位速度信号的曲线，两者之差才是稳态误差。单位速度信号的曲线 $r(t)=t$ 就是函数 $y=t$ 的曲线，很容易画出。

MATLAB 里也没有求等加速信号输入响应的函数，为了计算其稳态误差，按照求斜坡响应的办法，还可用 step ( ) 函数求其系统的响应。根据系统闭环传递函数的定义有：

$$\Phi(s)=\frac{C(s)}{R(s)}$$

对于单位等加速信号有：

$$r(t)=\frac{1}{2}a_0t^2\Big|_{a_0=1}=\frac{1}{2}t^2 \tag{4-32}$$

$$R(s)=\frac{1}{s^3} \tag{4-33}$$

$$C(s)=\Phi(s)\cdot R(s)=\Phi(s)\cdot\frac{1}{s^3}=\left[\Phi(s)\cdot\frac{1}{s^2}\right]\cdot\frac{1}{s} \tag{4-34}$$

由式(4-34）看出，可以将系统闭环传递函数除以拉氏算子“s^2”，再使用 step ( ) 函数求出的则是系统单位等加速输入信号的响应。在 MATLAB 程序中，这也只要在系统闭环传递函数的分母多项式乘以“s^2”即可。

为了计算系统的稳态误差，还需在绘制系统单位等加速输入信号响应曲线的同时，绘制出 $r(t)=\frac{1}{2}t^2$ 单位等加速信号的曲线，两者之差才是稳态误差。以下程序中特别考虑了单位等加速信号的 MATLAB 实现。需要指出的是在这个实现中，使用了单位冲激函数“impulse ( )”，其思考如下。考虑系统输出、传递函数、输入三者的关系有：

$$C(s)=\Phi(s)\cdot R(s)$$

假如把$\Phi(s)$看作系统输入，而将 $R(s)$看作系统传递函数，又若$\varphi(t)=\delta(t)$，$\Phi(s)=1$。在这个单位冲激函数作用下，则有：

$$C(s)=R(s)$$

所以，单位冲激响应

$$k(t)=c(t)=L^{-1}[C(s)]=L^{-1}[R(s)]=r(t)=\frac{1}{2}t^2 \tag{4-35}$$

4.3.2.2 三种典型信号输入作用下的稳态误差系数与稳态误差

三种典型信号输入作用下的系统型别、稳态误差系数及稳态误差与输入信号之间的关系列于表 4-6 中。这个表的结论对于计算系统稳态误差时的正确与错误判断，是非常重要、非常有用的。

表 4-6 系统输入信号作用下的稳态误差

型别	K_p	K_v	K_a	$r(t)=r_0$	$r(t)=v_0t$	$r(t)=a_0t^2/2$
0	K	0	0	$r_0/(1+K)$	∞	∞
I	∞	K	0	0	v_0/K	∞
II	∞	∞	K	0	0	a_0/K
III	∞	∞	∞	0	0	0

4.3.3 控制系统稳态误差计算举例

【例 4-16】已知一个单位负反馈系统开环零极点增益模型为:

$$G(s)=3\frac{(2s+1)}{s(s+2)(s-1)}$$

试绘制出该系统的单位斜坡响应曲线并求单位斜坡响应稳态误差。

【解】

（1）对系统判稳

根据题目已知条件给出的系统：

$$G(s)=3\frac{(2s+1)}{s(s+2)(s-1)}$$

此为系统的零极点增益模型。根据题意，调用函数 roots ( ) 命令的程序 L4301a.m 如下。

```
%MATLAB PROGRAM L4301a.m
%
k=6;z=-0.5;p=[-2 1 0];
[n1,d1]=zp2tf(z,p,k);
s=tf(n1,d1);
sys=feedback(s,1);
roots(sys.den{1})
ans =
  -0.1084 + 1.9541i
  -0.1084 - 1.9541i
  -0.7832
```

程序运行后所得系统闭环全部特征根的实部都是负值，说明闭环系统稳定。以下进行稳态误差的计算确是有意义的。

（2）求系统单位阶跃给定响应与稳态误差

根据题意，调用函数 step ( ) 命令的程序 L4301b.m 如下。该程序稍加整理完善可以改编为对于系统求其单位阶跃给定响应与稳态误差的函数 ster.( )，参见本节后的补充。

```
%MATLAB PROGRAM L4301b.m
%
k=6;z=-0.5;p=[-2 1 0];
[n1,d1]=zp2tf(z,p,k);
s=tf(n1,d1);
sys=feedback(s,1);
step(sys);
t=[0:0.1:300]';
y=step(sys,t);
subplot(121),plot(t,y),grid
subplot(122),ess=1-y;
plot(t,ess),grid
ess(length(ess))
```

该程序运行后可得系统的单位阶跃给定响应与误差响应曲线，如图 4-13 所示。

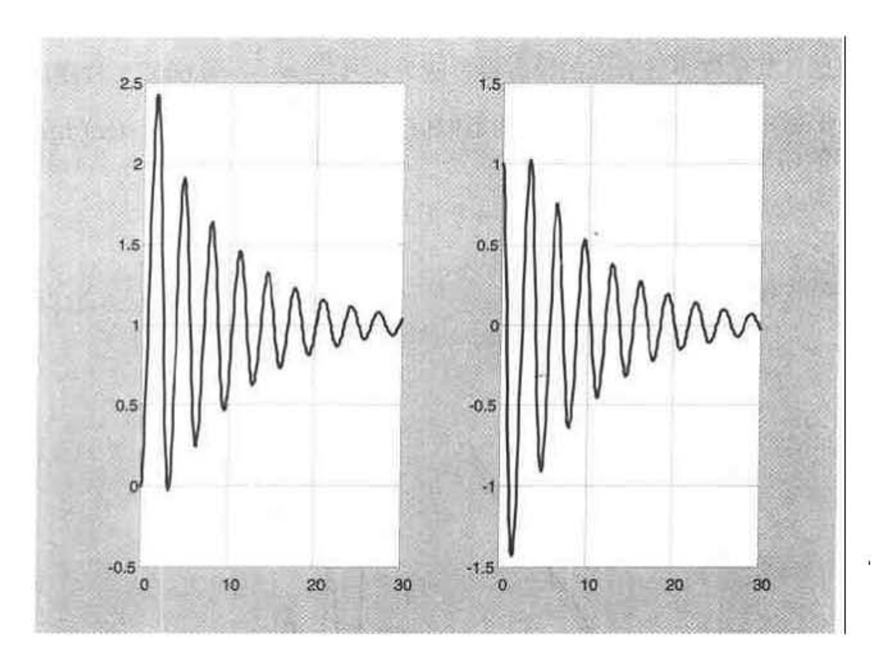


图 4-13 系统阶跃跃响应与阶跃误差响应曲线

根据表 4-6 得知，原系统为 I 型系统，其单位阶跃响应的稳态误差应为 $e_{ss}=0$，程序运行计算出结果(时间取为 300s)：*ess*=-4.6629e-015（实际为 0）。这已证明控制理论的正确。如果运行以下调用 ster.m 函数的程序，会得到同样的结果。

```
k=6;z=-0.5;p=[-2 1 0];[n1,d1]=zp2tf(z,p,k);
s=tf(n1,d1);
sys=feedback(s,1);
t=[0:0.1:300]';[ess]-ster(0,sys,t);
```

（3）求系统单位斜坡给定响应与稳态误差

根据以上介绍的计算斜坡响应稳态误差的思考与题目要求，调用函数命令 step ( ) 的程序 L4301c.m 如下。该程序稍加整理完善可以改编为对于系统求其单位斜坡给定响应与稳态误差的函数 ster.m。参见本节后附录。

```
%MATLAB PROGRAM L4301c.m
%
k=6;z=-0.5;p=[-2 1 0];[n1,d1]=zp2tf(z,p,k);
s=tf(n1,d1);sys=feedback(s,1);t=[0:0.1:50]';
num=sys.num{1};den=[sys.den{1},0];
sys=tf(num,den);y=step(sys,t);
subplot(121),plot(t,[t y]),grid
subplot(122),es=t-y;
plot(t,es),grid
ess=es(length(es))
```

执行程序后，可得系统的单位斜坡响应曲线，如图 4-14 所示。根据表 4-6 得知，原系

统为 I 型系统，其单位斜坡响应的稳态误差应为 $e_{ss}=\frac{u_0}{K}=\frac{1}{1.5}=0.6667$，程序运行计算出结果 ess = - 0.6678（这是时间取为 50s 的近似值）。如果运行以下调用 ster.( )函数的程序，会得到同样的结果。

```
k=6;z=-0.5;p=[-2 1 0];[n1,d1]=zp2tf(z,p,k);
s=tf(n1,d1);sys=feedback(s,1);t=[0:0.1:50]';
[ess1]=ster(1,sys,t);
```

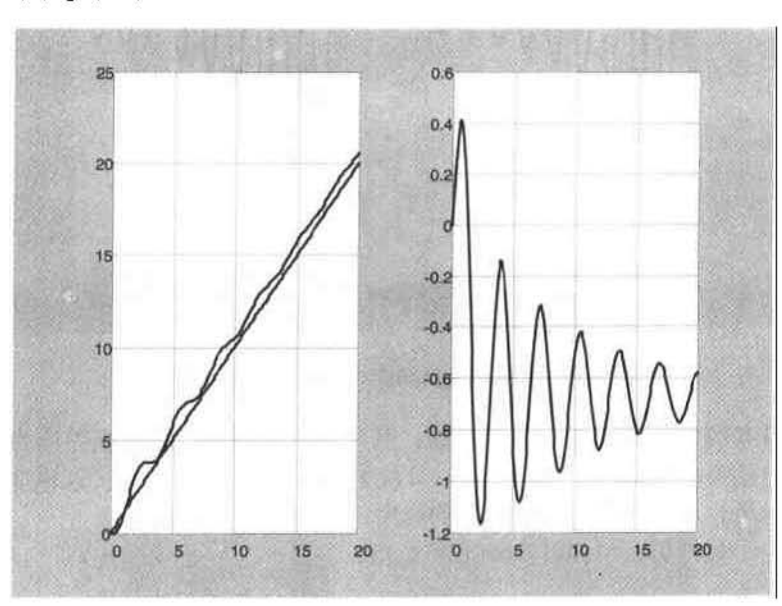


图 4-14 单位斜坡响应与单位斜坡误差响应曲线

【例 4-17】已知一个单位负反馈系统前向通道的传递函数为：

$$G(s)=\frac{24s^2+18s+3}{s^4+2s^3+10s^2}$$

试绘制出该系统的单位等加速信号输入响应及其稳态误差响应曲线，并计算其响应的稳态误差。

【解】

（1）对系统判稳

根据题目已知条件给出的系统：

$$G(s)=\frac{24s^2+18s+3}{s^4+2s^3+10s^2}$$

此为系统的传递函数模型。根据题意，调用函数 roots ( ) 命令的程序 L4302a.m 如下。

```
%MATLAB PROGRAM L4302a.m
%
n1=[24 18 3];d1=[1 2 10 0 0];
s1=tf(n1,d1);sys=feedback(s1,1);
```

```
roots(sys.den{1})
ans =
  -0.7303 + 5.7086i
  -0.7303 - 5.7086i
  -0.2697 + 0.1335i
  -0.2697 - 0.1335i
```

程序运行后所得系统闭环全部特征根的实部都是负值，说明闭环系统稳定。

（2）求系统单位等加速信号输入响应与其误差响应

根据题目要求，调用 step ( ) 及其相关函数命令的程序 L4302b.m 如下。该程序稍加整理完善可以改编为对于系统求其单位等加速信号输入响应与其误差的函数 ster.( )，参见本节后的附录。

```
%MATLAB PROGRAM L4302b.m
%
n1=3*conv([2 1],[4 1]);d1=conv([1 0 0],[1 2 10]);s1=tf(n1,d1);
sys=feedback(s1,1);t=[0:0.1:30]';
num1=sys.num{1};den1=[sys.den{1},0,0];sy1=tf(num1,den1);
y1=step(sy1,t);nu2=1;den2=[1 0 0 0];
sy2=tf(nu2,den2);y2=impulse(sy2,t);
subplot(121),plot(t,[y2 y1]),grid
subplot(122),es=y2-y1;plot(t,es),grid
ess=es(length(es));
```

执行程序后，可得系统单位等加速输入信号的响应曲线，如图 4-15 所示。

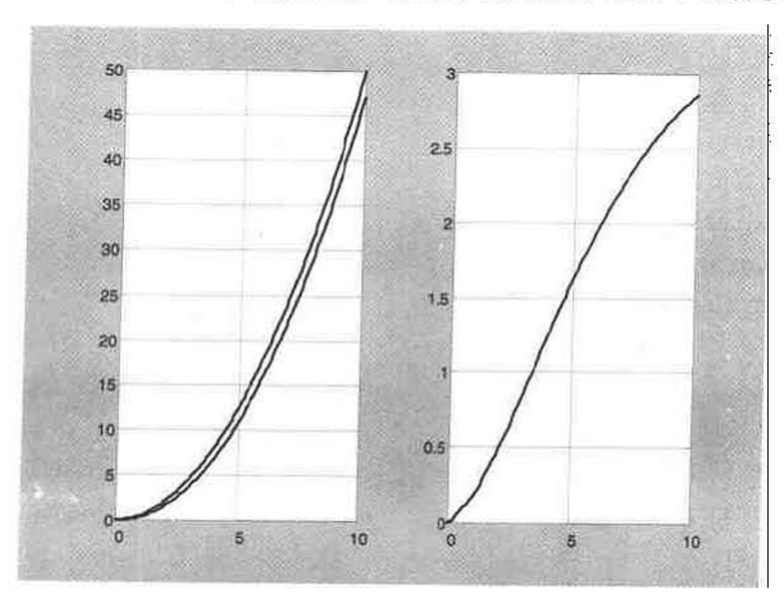


图 4-15　单位斜坡响应与单位斜坡误差响应曲线

根据表 4-6 得知，由题目给定的系统为Ⅱ型系统，其等加速输入信号响应的稳态误差应为 $e_{ss}=\dfrac{u_0}{K}=\dfrac{1}{0.3}=3.3333$，程序运行结果 *ess*=3.3335（这是时间取为 30s 的近似值）。如果运行以下调用 ster.( )函数的程序，会得到同样的结果。

```
n1=3*conv([2 1],[4 1]);d1=conv([1 0 0],[1 2 10]);
s1=tf(n1,d1);sys=feedback(s1,1);
t=[0:0.1:30]';[ess]=ster(2,sys,t);
```

附录：

【MATLAB PROGRAM 1】求系统单位阶跃给定响应与稳态误差、单位斜坡给定响应与稳态误差、单位等加速信号输入响应与其误差响应的函数 ster.( )

ster.( ) 函数的调用格式为：

[*ess*]=***ster***(*key*,*sys*,*t*)

其中，输入参数 *sys*，*t* 是系统闭环对应的传递函数与其计算响应的时间，还需根据系统的具体情况设置一个合适的时间向量（例如 *t*=[0:0.1:50]'）以求得准确的稳态误差。须特别注意时间向量的格式，必须要将行向量加一撇进行转置。*key* 参量取为 0 或 1 或 2，当 *key*=0 时，为求系统单位阶跃给定响应与稳态误差；当 *key*=1 时，为求单位斜坡给定响应与稳态误差；当 *key*=2 时，为求单位等加速信号输入响应与其误差响应函数。返回的是相应响应的稳态误差 *ess*。函数被调用后还同时绘制出相应响应曲线与误差响应曲线。

```
function [ess]=ster(key,sys,t)
% MATLAB PROGRAM ster.m
%
%Count ess0
if key==0
  y=step(sys,t);
  subplot(121),plot(t,y),grid
  subplot(122),es=1-y;
  plot(t,es),grid
  ess=es(length(es))
%Count ess1
elseif key==1
  num=sys.num{1};
  den=[sys.den{1},0];
  sys=tf(num,den);
  y=step(sys,t);
  subplot(121),plot(t,[t y]),grid
  subplot(122),es=t-y;
```

```
  plot(t,es),grid
  ess=es(length(es));
%count ess2
elseif key==2
  num1=sys.num{1};
  den1=[sys.den{1},0,0];
  sy1=tf(num1,den1);
  y1=step(sy1,t);
  nu2=1;den2=[1 0 0 0];
  sy2=tf(nu2,den2);
  y2=impulse(sy2,t);
  subplot(121),plot(t,[y2 y1]),grid
  subplot(122),es=y2-y1;
  plot(t,es),grid
  ess=es(length(es))
end
```

4.4 控制系统时域分析的 MATLAB 实现

4.4.1 自动控制的一些基本概念

为了讨论自动控制的仿真，首先简要回顾有关自动控制的一些基本概念。

1．自动控制

利用控制装置自动地操纵受控对象（机器设备或生产过程），使其具有给定的状态或性能，这就叫做自动控制。

2．自动控制系统

能够完成自动控制任务的系统叫做自动控制系统，它由控制器与受控对象组成。要求进行自动控制的机器、设备或生产过程是受控对象。对受控对象起控制作用的设备总体，叫控制器。

3．典型环节

构成控制系统的物理实体不同但数学模型相同的几种基本而简单的因子（环节）诸如惯性环节、比例环节、积分环节、微分环节、振荡环节等等，叫做典型环节。

4．传递函数

即线性定常系统在零初始条件下，输出量的拉氏变换与输入量的拉氏变换之比。

5．闭环系统的开环传递函数

即断开闭环反馈通道的输出通路，前向通道与反馈通道传递函数的乘积。

6．闭环系统的误差传递函数

系统给定信号与反馈通道的输出信号之差叫做系统的误差。系统的误差与给定输入

或扰动输入之间的传递函数叫做系统的误差传递函数。

7．系统的开环增益

是指系统开环传递函数中将各典型环节（各因式）的 s 零次方项（即 s^0）系数换算为 1 后的总比例系数。

8．控制系统的稳定性

即系统动态过程稳定的程度与系统受到扰动作用偏离原平衡状态，当去掉扰动后系统恢复平衡工作状态的能力。

9．控制系统的快速性

即系统动态过程经历时间长短的快慢程度，若动态过程时间短，则系统的快速性能好。

10．控制系统的准确性

即动态过程结束进入平衡工作状态后，系统被控量对给定值的偏差之大小程度。其偏差越小，即系统控制精度越高，系统的准确性越好。

11．系统的动态过程

即控制系统在受到给定信号或者干扰信号作用后，系统被控制量变化的全过程。

12．控制系统的性能指标

控制系统的性能指标有静态（或稳态）与动态两类指标。动态性能指标又分为跟随性能指标与抗扰动性能指标两种。跟随性能指标有上升时间 t_r、峰值时间 t_p、超调量 $\sigma\%$、调节时间 t_s 等；抗扰动性能指标有动态降落、恢复时间等。

13．典型外作用信号

是众多而复杂的实际外作用信号的近似与抽象。它的选择不仅应使数学运算简单，而且还应便于用实验验证。常用的典型外作用信号有：阶跃信号、斜坡信号、加速度信号、单位理想脉冲（即单位冲激信号）与正弦信号。

14．典型跟随过程

即以输出量的初始值为零、在给定阶跃信号作用下的过渡过程。这时系统的动态响应也称作阶跃响应。

15．典型时间响应

即初始状态为零的系统，在典型外作用信号下系统输出量的动态过程。在以上的典型外作用下，有阶跃响应、斜坡响应、单位冲激响应及正弦响应等。

16．阶跃响应性能指标

即控制系统在跟踪或复现阶跃输入信号时，响应过程的性能指标。因为阶跃信号对于系统是较为恶劣、严格的工作条件，所以常以阶跃响应衡量系统控制性能的好坏。

17．超调量 $\sigma\%$

指阶跃响应曲线 $h(t)$ 中对稳态值的最大超出量与稳态值之比。即

$$\sigma\% = \frac{h(t_p)-h(\infty)}{h(\infty)}\cdot 100\% \tag{4-36}$$

在二阶系统中，超调量 $\sigma\%$ 与阻尼比 ς 之间的关系为：

$$\sigma\% = e^{-\frac{\pi\varsigma}{\sqrt{1-\varsigma^2}}} \times 100\%$$

若是已知阻尼比ς，可以用以下指令求超调量$\sigma\%$：

```
sigma=2.7182^(-pi*zeta/(1-(zeta)^2)^(1/2))
```

以上问题的逆命题是已知系统超调量$\sigma\%$ (sigma)，则可用以下指令求阻尼比ς：

```
zeta=((log(1/sigma))^2/((pi)^2+(log(1/sigma))^2))^(1/2)
```

【例 4-18】已知二阶系统的$\varsigma = 0.46$，求系统的超调量$\sigma\%$。

【解】

执行以下指令即可求系统的超调量：

```
zeta=0.46;
sigma=2.7182^(-pi*zeta/(1-(zeta)^2)^(1/2))
sigma =
    0.1964
```

即求得系统的超调量$\sigma\%$=19.64%。

【例 4-19】已知二阶系统的超调量$\sigma\%$=20%，求系统的阻尼比ς。

【解】

已知系统超调量$\sigma\%$ (sgm)，只要输入以下 MATLAB 指令即可求系统的阻尼比ς。

```
sigma=0.2;
zeta=((log(1/sigma))^2/((pi)^2+(log(1/sigma))^2))^(1/2)
zeta =
    0.4559
```

即求得系统的阻尼比ς =0.4559。

18．峰值时间 t_p

指从 0 到阶跃响应曲线$h(t)$中超过其稳态值而达第一个峰值之间所需要的时间。

19．调节时间 t_s

指阶跃响应曲线中，$h(t)$进入稳态值附近$\pm 5\% h(\infty)$[或$\pm 2\% h(\infty)$]的误差带而不再超出的最小时间，t_s也称过度过程时间。

20．典型的抗扰过程

系统稳定运行中突加一个使输出量降低的负扰动以后的过程。

21．动态降落

系统稳定运行时，突加一个约定的标准负扰动作用，在过渡过程中所引起的输出量最大降落值ΔC_{max}。

22．恢复时间

从阶跃扰动作用开始到输出量基本上恢复稳态的过程中，输出量与新稳态值$C_{\infty 2}$之差进入某基准量C_b的±5%（或±2%）范围之内所需的时间。

4.4.2 控制系统时域响应仿真的主要问题

时域分析法是根据自动控制系统微分方程，用拉普拉斯变换求解动态响应的过程曲线。典型的动态过程响应有单位阶跃响应、单位斜坡响应、单位加速度响应与单位冲激响应等。

时域分析的另外一个目的是求解响应的性能指标。通常将控制系统跟踪或复现阶跃输入信号响应的指标作为系统控制性能的指标。阶跃响应的一般性能指标有：峰值时间 t_p、超调量 sigma($\sigma\%$)、调节时间 t_s 及稳态误差 e_{ss}。

响应曲线是一个以时间为自变量的过程曲线。最原始的方法是设置多个时间点，计算出曲线上各个点的函数值，逐点连接描绘成线，甚至要使用曲线板，须经过多次坐标点的繁琐计算，画出的曲线不仅有误差，而且很是麻烦而不方便。

计算机仿真以及本书论述的 MATLAB 仿真是用先进的计算机程序来实现动态响应曲线的绘制过程，既简单方便，又快捷高效而准确。

此外，MATLAB 仿真绘制的响应曲线，纵、横坐标都自动产生有自适应能力的精确刻度，从曲线的形状可以直接计算出响应的性能指标。

4.4.3 时域分析 MATLAB 实现的方法

与控制系统其他的 MATLAB 仿真一样，时域响应 MATLAB 仿真的方法也有两种：一种是在 MATLAB 的函数指令方式下进行时域仿真；另一种是在 SIMULINK 窗口菜单操作方式下进行时域仿真。

已经指出过，本书所有例题中给出的程序只要不是必须在 MATLAB 命令窗中运行的，均是在 Notebook 中运行并绘制出曲线与图形，而不必退出 WORD 转换到 MATLAB 中运行程序。

4.4.4 MATLAB 函数指令方式下的时域响应仿真

利用 MATLAB 所提供的求取连续系统的单位阶跃响应函数 step（），单位冲激响应函数 impulse（），零输入响应函数 initial（） 以及其他函数都可以求出其相应的响应。

MATLAB 也提供了相应离散系统的单位阶跃响应函数 dstep（），单位冲激响应函数 dimpulse（），零输入响应函数 dinitial（）以及其他函数也可以求出其相应的响应。

单位阶跃响应是系统在单位阶跃信号 1（t）作用下的响应。工程上通常用阶跃响应的性能指标作为控制系统性能的指标。在 MATLAB 里用 step（） 与 dstep（） 函数来求其响应曲线。

单位冲激响应是系统在单位脉冲信号（或单位冲激信号）$\delta(t)$ 作用下的响应。单位脉冲信号 $\delta(t)$ 是一个重要的数学工具。理想的 $\delta(t)$ 在现实中虽不存在，但实际工程中的任何突加的作用力、脉动电压信号、冲击力、阵风或大气流，都可近似作为 $\delta(t)$。在 MATLAB 里用 impulse（）与 dimpulse（） 函数来求其响应曲线。

零输入响应是系统全响应的自由分量，此分量是系统微分方程对应的齐次方程的通解，它与系统的外作用无关，只与系统的结构参数及初始状态有关。工程上使用也较多。

在 MATLAB 里用 initial（）与 dinitial（）函数来求其响应曲线。

应用 MATLAB 提供的上述时域仿真函数与其他函数命令、语句，编制成程序，这种 MATLAB 的指令方式下进行时域仿真是最常用的实现方法。

4.4.4.1　时域响应 MATLAB 仿真的函数指令格式

1．求连续系统单位阶跃响应的函数 step ( )

函数命令调用格式如下：

[*y*, ***t***, ***x***] = **step** (*sys*)

step (*sys*)

step (*sys*, *t*)

step (*sys*1,*sys*2,…,*sysN*)

step (*sys*1,*sys*2,…,*sysN*,*t*)

step(*sys*1,'*PlotStyle*1',…*sysN*,'*PlotStyleN* ')

函数命令使用说明：

step (*sys*) 函数用来计算系统的单位阶跃响应，可用于 SISO 或者 MIMO 的连续时间系统或者离散时间系统。计算离散时间系统阶跃响应的函数为 dstep ()。当函数命令为无等式左边输出变量的格式时，函数在当前图形窗口中直接绘制出系统的阶跃响应曲线。LTI 对象 sys 可以是由函数 tf ( )、zpk ( )、ss ( ) 中任何一个建立的系统模型。

step (*sys*, *t*) 函数用于计算系统的阶跃响应，函数中 *t* 可以指定为一个仿真终止时间，此时 t 为一标量；也可以设置为一个时间矢量（例如用 t=0:dt:Tfinal 命令）。若是离散系统，时间间隔 dt 必须与采样周期匹配。

step (*sys*1,*sys*2,…,*sysn*) 函数可同时仿真多个 LTI 对象。

step(*sys*1,'*PlotStyle*1',…*sysn*,'*PlotStylen*') 函数可在仿真中设置仿真绘制曲线的属性，其 *PlotStyle* 应是 MATLAB 标准函数命令 plot（） 支持的各种属性标识符字符串。

[*y*, ***t***, ***x***] = step (*sys*) 函数为带有输出变量引用的函数，可计算系统阶跃响应的输出数据，而不绘制出曲线。输出变量 ***y*** 是系统的输出响应矢量；输出变量 ***t*** 为取积分值的时间矢量；输出变量 ***x*** 是系统的状态轨迹数据。对于 MIMO 系统，***y*** 的维数为：(length of t)*(number of outputs)*(number of inputs)，y(:,:,j)，对应于第 *j* 个输入通道；同此，***x*** 的维数为：(length of t)*(number of states)*(number of inputs)。以下函数命令同此。

需要特别指出的是，输出变量[*y*, *t*, *x*] 三个元素的顺序不能错，尤其是 ***t*** 与 ***x*** 的顺序，在 MATLAB 5.x 中的规定与过去的老版本正好相反。输出变量[***y***, ***t***, ***x***] 三个元素的顺序如果弄错，会给与 step ( ) 函数的相关计算带来错误。在阅读或修改老版本的 MATLAB 程序时这是需要特别注意的。

【例 4-20】已知单位负反馈系统前向通道的传递函数为：

$$G(s)=\frac{80}{s^2+2s}$$

试作出其单位阶跃响应曲线与误差响应曲线。

【解】

根据题目要求，用函数命令编写的程序 L4401.m 如下。

```
% MATLAB PROGRAM L4401.m
%
s1=tf(80,[1 2 0]);
closys=feedback(sys,1);
figure(1);
step(closys);hold on
t1=[0:5:20];[y,t]=step(closys);
figure(2);ess=1-y;
plot(t,ess);y1=step(closys,t1)
ess1=1-y1;
```

运行该程序可得系统的单位阶跃给定响应曲线（图 4-16）与误差响应曲线（图 4-17）。

【例 4-21】已知二阶系统传递函数为：

$$\Phi(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

当 $\omega_n=1$ 时，试计算当阻尼比 ς 值从 0.1 到 1 时二阶系统的阶跃响应，并且绘制一簇阶跃响应三维图。

【解】

根据题目要求，用函数命令编写如下程序 L4402.m。

```
%MATLAB PROGRAM L4402.m
%
num=1;y=zeros(200,1);i=0;
```

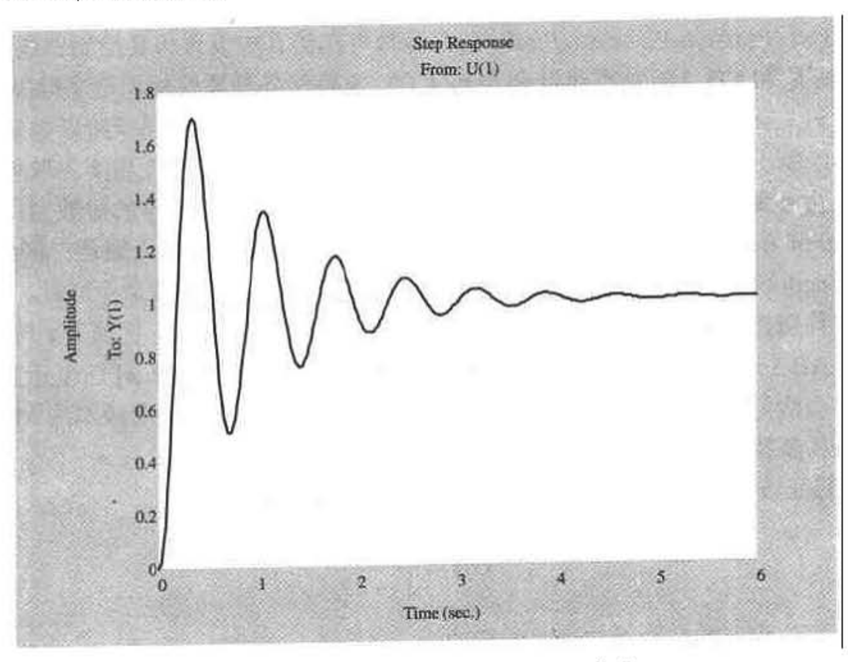


图 4-16　系统的单位阶跃给定响应曲线

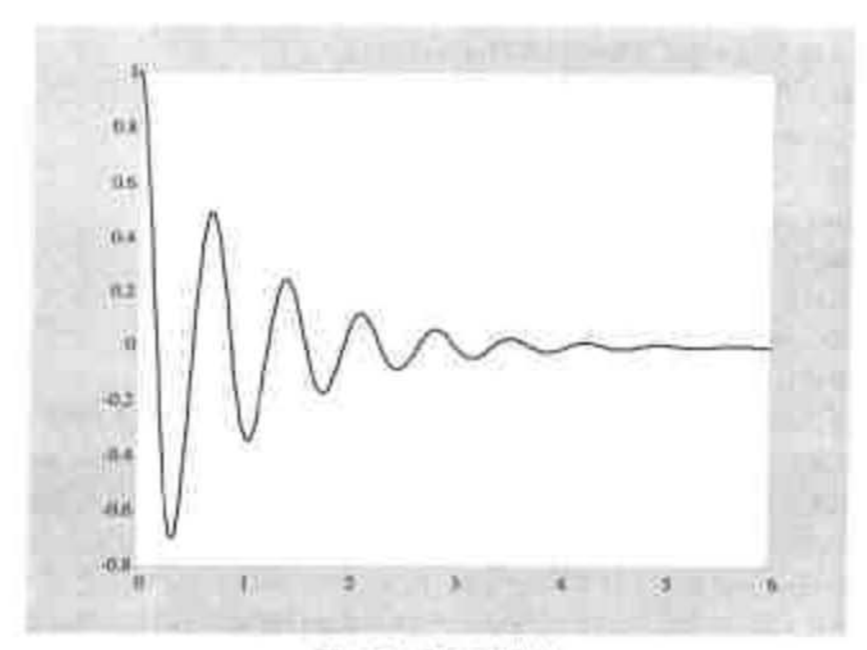


图 4-17 误差响应曲线

```
for bc=0.1:0.1:1
     den=[1,2*bc,1];t=[0:0.1:19.9]';sys=tf(num,den);
     i=i+1; y(:,i)=step(sys,t);
 end
mesh(flipud(y),[-100 20])
```

运行该程序得图 4-18 。

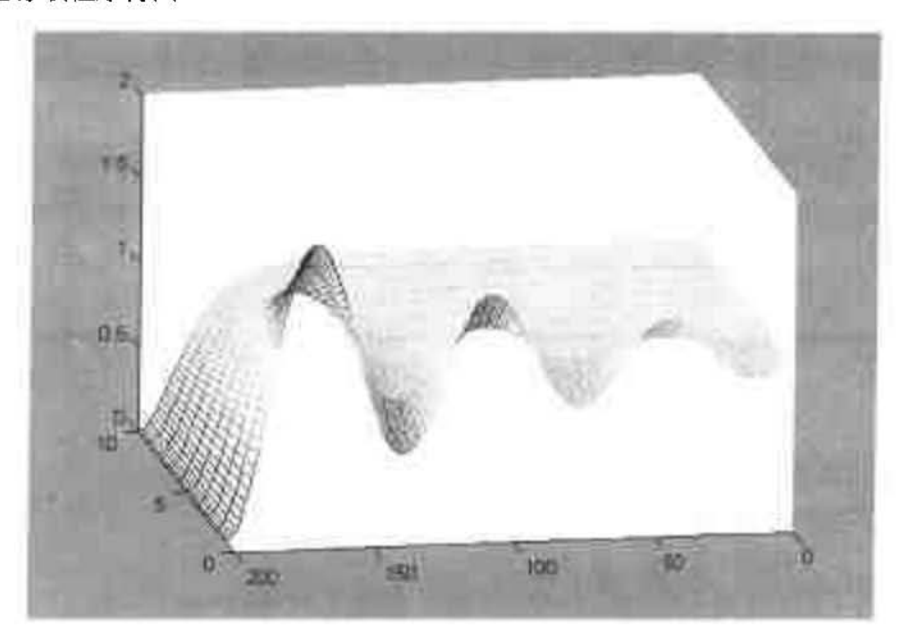


图 4-18 二阶系统阻尼比变化时的阶跃响应曲线簇

2. 求连续系统单位冲激响应的函数 impulse ( )

函数命令调用格式如下：

[*y*, *t*, *x*] = **impulse** (*sys*)

impulse (*sys*)

impulse (*sys*, t)

impulse (*sys*1,*sys*2,…,*sysN*)

impulse (*sys*1,*sys*2,…,*sysN*,*t*)

impulse (*sys*1,'*PlotStyle*1',…*sysN*,'*PlotStyleN*')

函数命令使用说明：

impulse (*sys*) 函数用来计算系统的单位冲激响应，可用于 SISO 或者 MIMO 的连续时间系统或者离散时间系统。计算离散时间系统单位冲激响应的函数为 dimpulse ( )。当函数命令为无等式左边输出变量的格式时，函数在当前图形窗口中直接绘制出系统的单位冲激响应曲线。LTI 对象 *sys* 可以是由函数 tf ( )、zpk ( )、ss ( ) 中任何一个建立的系统模型。

impulse (*sys*, *t*) 函数用于计算系统的单位冲激响应，函数中 t 可以指定为一个仿真终止时间，此时 *t* 为一标量；也可以设置为一个时间矢量（例如用 t=0:dt:Tfinal 命令）。若是离散系统，时间间隔 dt 必须与采样周期匹配。

impulse (*sys*1,*sys*2,…,*sysN*) 函数可同时仿真多个 LTI 对象。

impulse (*sys*1,'*PlotStyle*1',…*sysN*,'*PlotStyleN*') 函数可在仿真中设置仿真绘制曲线的属性，其 PlotStyle 应是 MATLAB 标准函数命令 plot ( ) 支持的各种属性标识符字符串。

[*y*, *t*, *x*] =impulse (*sys*) 函数为带有输出变量引用的函数，可计算系统单位冲激响应的输出数据，而不绘制出曲线。*y*、*t* 与 *x* 的其他解释同 step ( ) 函数的。

【例 4-22】求图 4-19 所示系统的单位冲激响应。

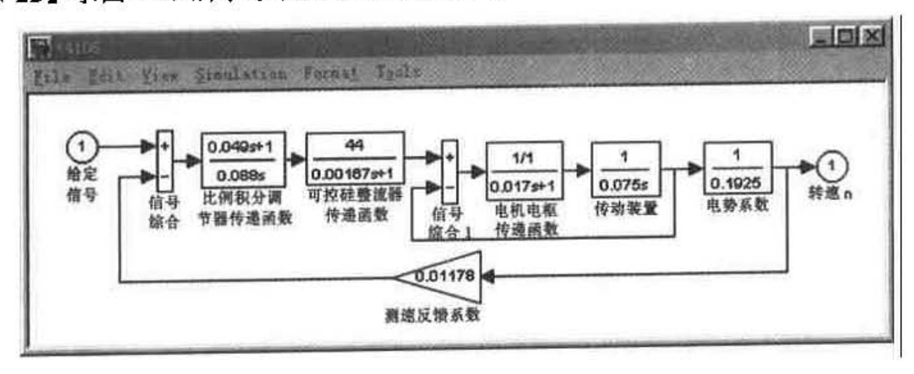


图 4-19 直流单闭环调速系统

【解】

根据题目要求，用 MATLAB 函数命令 impulse ( ) 编写程序 L4403.m。

```
%MATLAB PROGRAM L4403.m
%
n1=[1];d1=[0.017 1];s1=tf(n1,d1);n2=[1];d2=[0.075 0];s2=tf(n2,d2);
sys1=feedback(s1*s2,1);n3=[0.049 1];d3=[0.088 0];s3=tf(n3,d3);
n4=[0 44];d4=[0.00167 1];s4=tf(n4,d4);
```

```
n5=1;d5=0.1925;s5=tf(n5,d5);n6=0.01178;d6=1;s6=tf(n6,d6);
sysq=sys1*s3*s4*s5;sys=feedback(sysq,s6);impulse(sys)
```

运行该程序可得如图 4-20 所示的单闭环调速系统单位冲激响应曲线。

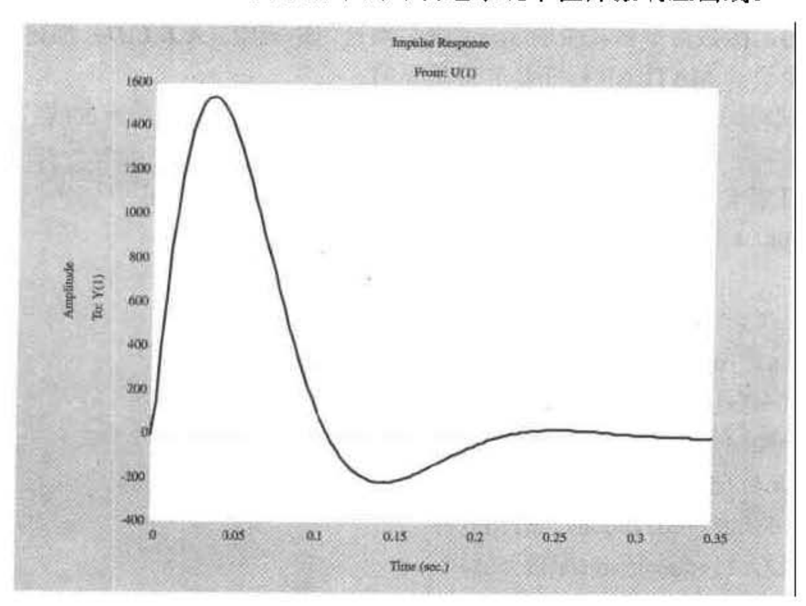


图 4-20 系统的单位冲激响应曲线

3. 求连续系统零输入响应的函数 initial ( )

函数命令调用格式如下:

[*y*, *t*, *x*] = initial (*sys*,*x*0)

initial (*sys*,*x*0)

initial (*sys*, *x*0,*t*)

initial(*sys*1,*sys*2,…,*sysN*,*x*0)

initial (*sys*1,*sys*2,…,*sysN*,*x*0,*t*)

initial(*sys*1,'*PlotStyle*1',…*sysN*,'*PlotStyleN*',*x*0)

函数命令使用说明:

initial (*sys*,*x*0) 函数用来计算系统的零输入响应，*x*0 为初始状态。需要特别指出，*sys* 系统必须是由矩阵组(***A***,***B***,***C***,***D***)描述的状态空间模型。该函数可用于 SISO 或者 MIMO 的连续时间系统或者离散时间系统。计算离散时间系统零输入响应的函数为 dinitial()。当函数命令为无等式左边输出变量的格式时，函数在当前图形窗口中直接绘制出系统的零输入响应曲线。

initial (*sys*,*x*0, *t*) 函数用于计算系统的零输入响应，函数中 *t* 可以指定为一个仿真终止时间，此时 *t* 为一标量；也可以设置为一个时间矢量（例如用 t=0:dt:Tfinal 命令）。若是离散系统，时间间隔 d *t* 必须与采样周期匹配。

initial (*sys*1,*sys*2,…,*sysN*,*x*0) 函数可同时仿真多个 LTI 对象。

initial (*sys*1,'*PlotStyle*1',…*sysN*,'*PlotStylen*',*x*0) 函数可在仿真中设置仿真绘制曲线的属性，其 PlotStyle1 应是 MATLAB 标准函数命令 plot（） 支持的各种属性标识符字符串。

$[y, t, x]$ =initial (*sys*,*x*0) 函数为带有输出变量引用的函数，可计算系统单位冲激响应的输出数据，而不绘制出曲线。输出变量 y 是系统的输出响应矢量；输出变量 t 为取积分值的时间矢量；输出变量 x 是系统的状态轨迹数据。

再次指出，函数命令格式中的 *sys* 系统必须是由矩阵组（***A***,***B***,***C***,***D***）描述的状态空间模型，这对于设计 MATLAB 程序是非常重要的。

【例 4-23】试对图 4-19 所示系统求其状态空间模型并绘制零输入响应曲线。

【解】

根据题目要求，用 **MATLAB** 函数命令编写如下程序 L4404.m。

```
%MATLAB PROGRAM L4404.m
%
n1=[0 1];d1=[0.017 1];n2=[0 1];d2=[0.075 0];
n12=conv(n1,n2);d12=conv(d1,d2);nn1=n12;dd1=n12+d12;
n3=[0.049 1];d3=[0.088 0];n4=[0 44];d4=[0.00167 1];
n5=[0 1];d5=[0 0.1925];n6=0.01178;d6=1;
nn2=conv(n3,n4);dd2=conv(d3,d4);
nn12=conv(nn1,nn2);dd12=conv(dd1,dd2);
num=conv(nn12,n5);den=conv(dd12,d5);
numc=num;denc=conv(n6,num)+conv(d6,den);
[a,b,c,d]=tf2ss(numc,denc)
x0=[1 0 0 0];t=0:0.01:0.5;
initial(a,b,c,d,x0,t)
```

运行该程序可得连续系统的状态空间模型与零输入响应曲线（图 4-21）。

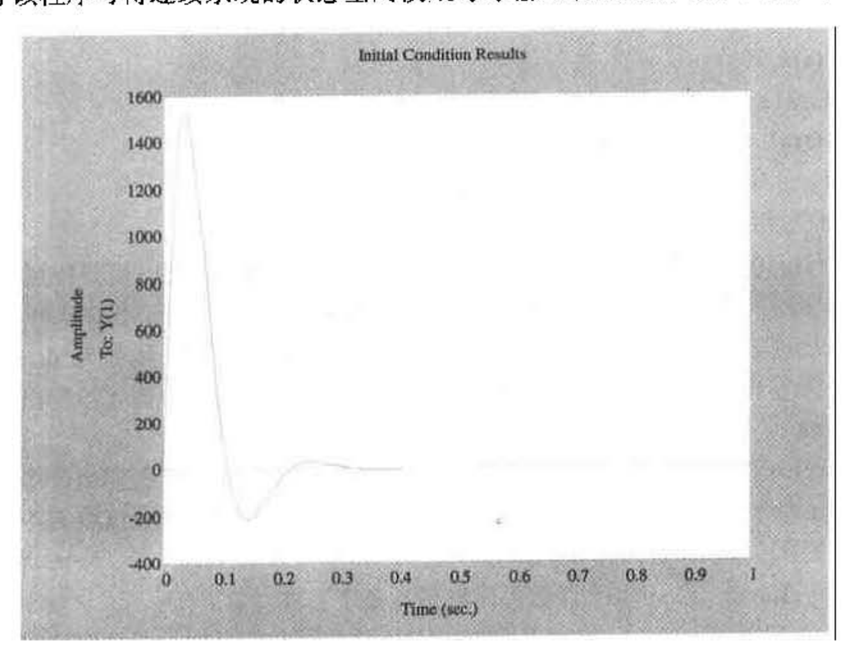


图 4-21　连续系统的零输入响应曲线

```
a =
   1.0e+007 *
    -0.0001   -0.0036   -0.1174   -1.4370
     0.0000         0         0         0
          0    0.0000         0         0
          0         0    0.0000         0
b =
     1
     0
     0
     0
c =
   1.0e+009 *
          0         0    0.0598    1.2199
d =
     0
```

4.4.4.2 MATLAB 函数指令时域响应仿真实例

【例 4-24】设控制系统的开环传递函数为：

$$G(s)=\frac{1.25}{s^2+s}$$

试绘制出该闭环系统的单位响应曲线，并计算系统的性能指标。

【解】

这个问题已知的系统数学模型是传递函数模型，用该种模型解决这一问题的程序为 L4405a.m。该程序稍加整理完善可以改编为对于阶跃响应求其多项性能指标的 MATLAB 函数 targ.( )，参见本节后附录。

```
% MATLAB PROGRAM L4405a.m
%
global y t
sys=tf(1.25,[1 1 0]);
Gc=feedback(sys,1);
step(Gc)
[y,t]=step(Gc);
[mp,tf]=max(y);
ct=length(t);
tm=max(t);
yss=y(ct);
q=1;
m=q-1;
```

```
while m<3,
  for a=(tm/100):0.01:tm
    j=[0:a:tm];
      for i=1:length(j);
        if (y(i+1)-y(i))<0 & (y(i)-y(i-1))>0,
            m=m+1;
            pm(m)=y(i);
            tp(m)=t(i);
        end
      end
  end
end
yss=y(ct);
ess=1-yss
b1=pm(1)-yss
b2=pm(2)-yss
sigma =100*b1/yss
n=b1/b2
pusi=(b1-b2)/b1
T=(tp(2)-tp(1))
f=1/T
```

程序执行后，即可得到如图 4-22 所示的单位阶跃响应曲线，并计算出了系统的性能指标如下：

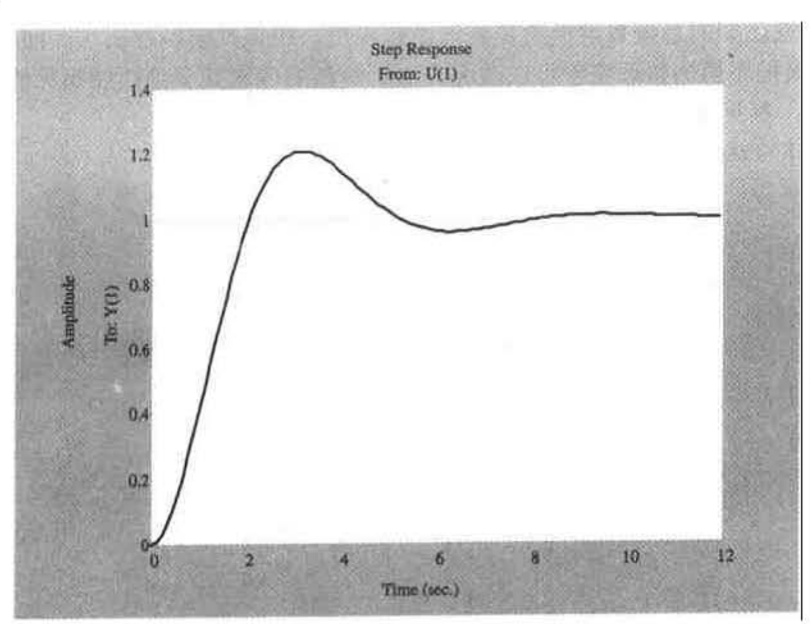


图 4-22　连续系统的阶跃响应曲线

阶跃响应的余差：*ess* = 0.0013

阶跃响应的第一（正向）波峰值：*b*1 = 0.2088

阶跃响应的第二（正向）波峰值：*b*2 = 0.0103

阶跃响应的超调量：*sigma* = 20.9121

阶跃响应的衰减比：*n* = 20.3364

阶跃响应的衰减率：*pusi* = 0.9508

阶跃响应的衰减振荡周期：*T* = 6.2945

阶跃响应的振荡频率：*f* = 0.1589

解决这一问题的另一个程序为 L4405b.m，该程序稍加整理完善可以改编为对于阶跃响应求其性能指标的 MATLAB 函数 perf.( )。该函数可以求系统单位阶跃响应超调量、峰值时间、调节时间，调节时间，可以选择 5% 或者 2% 的误差带，参见本节后的附录。

```
% MATLAB PROGRAM L4405b.m
%
n1=[1.25];d1=[1 1 0];s1=tf(n1,d1);
Gc=feedback(s1,1);
step(Gc);
[y,t]=step(Gc);
%Count Sigma and tp
[mp,tf]=max(y);
cs=length(t);
yss=y(cs);
Sigma =100*(mp-yss)/yss
tp=t(tf)
%Count ts
i=cs+1;
n=0;
while n==0,
        i=i-1;
            if i==1,
               n=1;
            elseif y(i)>1.05*yss,
               n=1;
            end;
end;
t1=t(i);
cs=length(t);
j=cs+1;
n=0;
```

```
while n==0,
      j=j-1;
          if j==1,
            n=1;
          elseif y(j)<0.95*yss,
n=1;
          end;
end;
t2=t(j);
      if t2<tp
         if t1>t2
            ts=t1
         end
      elseif t2>tp,
         if t2<t1,
            ts=t2
         else
            ts=t1
         end
      end
```

程序执行后，仍得到如图 4-22 所示的单位阶跃响应曲线，并计算出了系统的性能指标如下：

超调量：*sigma* = 20.9121 %

峰值时间：*tp*= 3.0920 s

调节时间：*ts* = 4.6380 s

程序运行求得的是 5 % 误差带调节时间，调用函数 perf.() 且当选择 key=2 时，求得的是 2 % 误差带调节时间。给出以下程序 L4405c.m。

```
% MATLAB PROGRAM L4405c.m
%
global y t
s1=tf(1.25,[1 1 0]);
sys=feedback(s1,1);
step(sys);
[y,t]=step(sys);
perf(2,y,t);
```

程序运行可相应求得 2 %误差带调节时间：

超调量：*sigma* = 20.9121 %

峰值时间：*tp*= 3.0920 s

调节时间：*ts* = 4.9693 s

有了单位阶跃响应曲线与性能指标，按自动控制理论，就可对该系统的阶跃响应性能等进行各种分析。

【例 4-25】已知一个二阶系统为：

$$G(s)=\frac{k}{s^2+cs+k}$$

$$c=\{1,2,4\},\quad k=\{1.25,2,29\}$$

试绘制出该系统所对应的三组不同参数配合下的阶跃响应三维图及同一坐标轴里的三条阶跃响应曲线。

【解】

由已知的系统传递函数数学模型，并根据题目要求，编制如下程序 L4406a.m。

```
%MATLAB PROGRAM L4406a.m
%
c=[1 2 4];k=[1.25 2 29];y=zeros(100,3);t=linspace(0,10,100)';
for j=1:3
    num=k(j);
    den=[1 c(j) k(j)];
    sys=tf(num,den);
    y(:,j)=step(sys,t);
end
mesh(flipud(y),[-100 20])
```

程序执行后，即可得到如图 4-23 所示的单位阶跃响应曲线三维图。

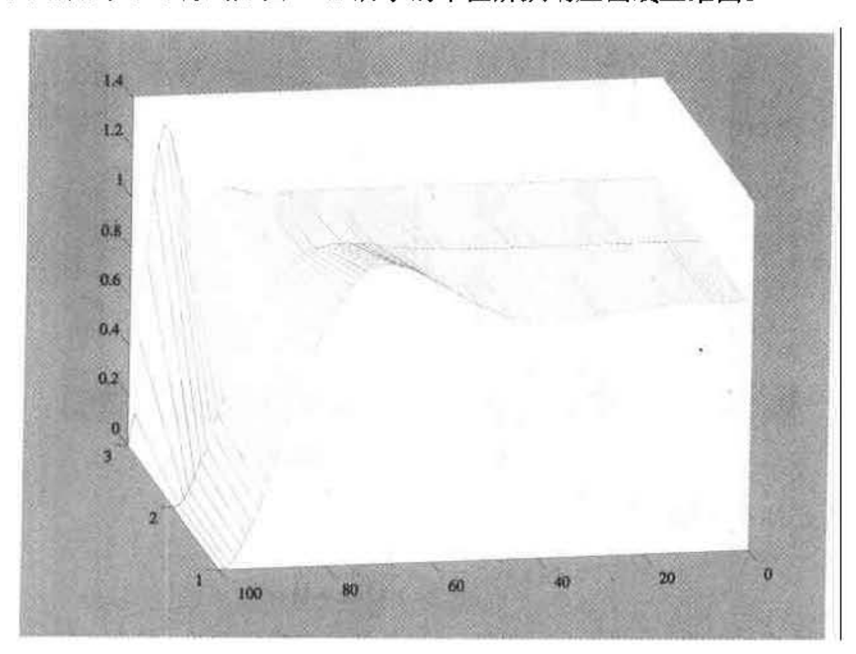


图 4-23　二阶系统的单位阶跃响应曲线三维图

为在同一坐标轴里绘制三条阶跃响应曲线并标注其参数，再编制程序 L4406b.m 如下：

```
%MATLAB PROGRAM L4406b.m
%
c=[1 2 4];k=[1.25 2 29];
t=linspace(0,10,100)';
for j=1:3
   num=k(j);den=[1 c(j) k(j)];
   sys=tf(num,den);
   y(:,j)=step(sys,t);
end
plot(t,y(:,1:3)),grid
gtext('a=1 b=1.25'),
gtext('a=2 b=2'),
gtext('a=4 b=29')
```

程序执行后，有如图 4-24 所示的标注有其对应参数的三条单位阶跃响应曲线。由曲线可知，第一组参数的系统振荡超调近 30%；第二组参数的系统超调近 20%；第三组参数的系统略微超调近 5%。

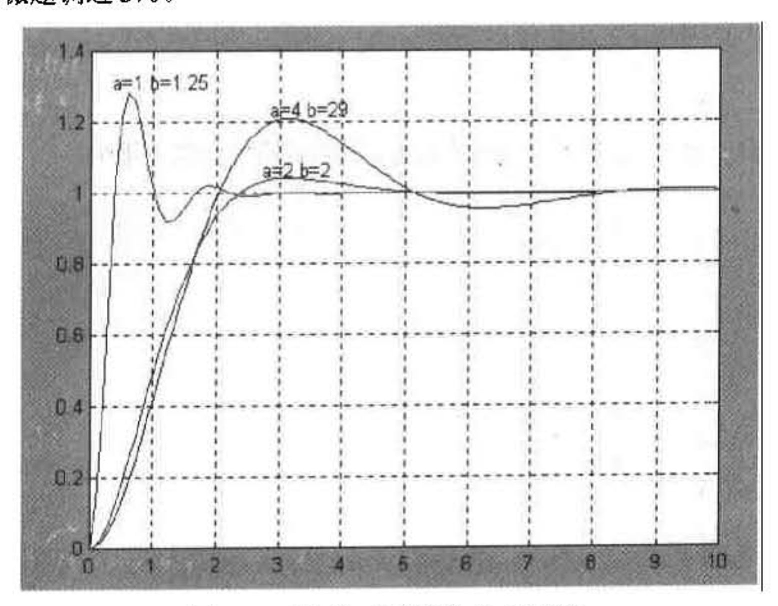


图 4-24 系统的三条单位阶跃响应曲线

【例 4-26】已知一个单位负反馈系统为：

$$G(s)=\frac{k}{s(0.5s+1)(4s+1)}$$

试绘制该系统当 k 分别为 1.4, 2.3, 3.5 时的单位阶跃给定响应曲线（绘制在同一张图上），并求当 k=1.4 时的单位阶跃给定响应性能指标。

【解】

根据题目要求，用函数命令编写 MATLAB 程序 L4407a.m。

```
% MATLAB PROGRAM L4407a.m
%
num=1;den=conv(conv([1 0],[0.5 1]),[4 1]);
rangek=[1.4 2.3 3.5];
t=linspace(0,20,200)';
for j=1:3
   s1=tf(num*rangek(j),den);
   sys=feedback(s1,1);
   y(:,j)=step(sys,t);
end
plot(t,y(:,1:3)),grid
gtext('k=1.4'),gtext('k=2.3'),
gtext('k=3.5')
```

执行程序后，有如图 4-25 所示的标注有其对应参数的三条单位阶跃响应曲线。由曲线可以看出，当 k=1.4 时，阶跃响应衰减振荡，系统稳定；当 k=2.25 时，阶跃响应等幅振荡，系统临界稳定；当 k=3.5 时，阶跃响应振荡发散，系统不稳定。

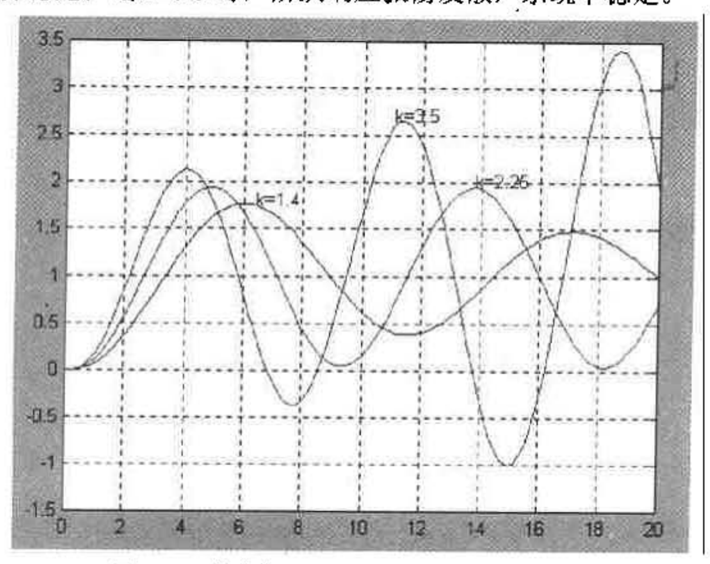


图 4-25　单位负反馈系统的三条阶跃响应曲线

为了求当 k=1.4 时单位阶跃给定响应(2%误差带时)的性能指标，给出以下程序 L4407b.m。

```
%MATLAB PROGRAM L4407b.m
%
global y t sys
n1=1.4;d1=conv(conv([1 0],[0.5 1]),[4 1]);
```

```
s1=tf(n1,d1);sys=feedback(s1,1);
step(sys);[y,t]=step(sys);
[sigma,tp,ts]=perf(2,y,t);
[mp,tp,b1,b2,n,pusi,tf]=targ(y,t);
```

执行程序后，可得如图 4-26 所示的单位阶跃给定响应曲线。

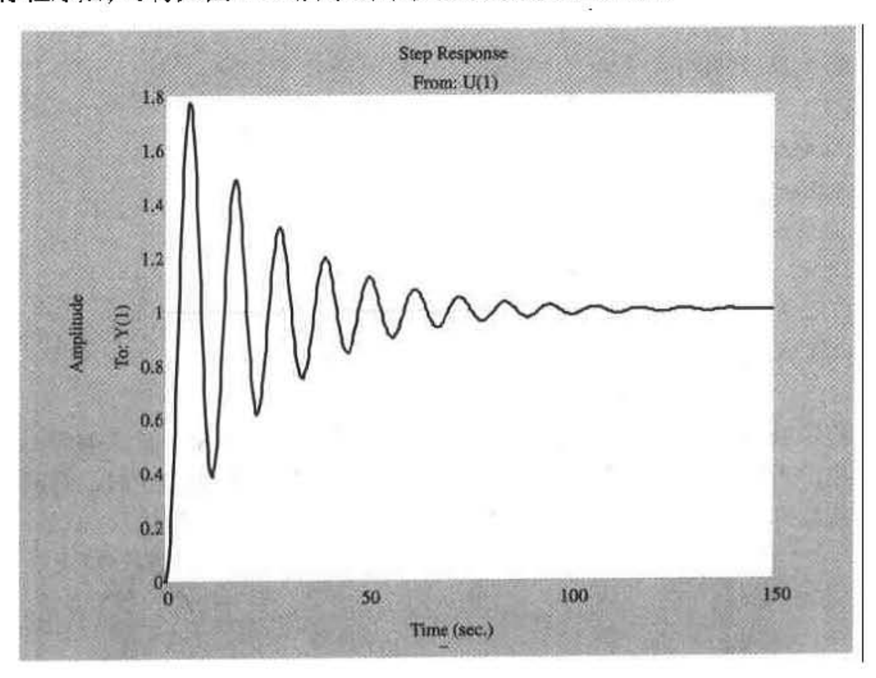


图 4-26 k=1.4 时系统单位阶跃给定响应曲线

并计算出阶跃给定响应性能指标（指标含义同上）如下：

sigma = 76.659 %

tp = 5.9467 s

ts = 84.876 s

ess = -0.0021

*b*1 = 0.7682

*b*2 = 0.4876

n = 1.5755

pusi = 0.3653

T = 11.0825

f = 0.0902

4.4.5 利用 SIMULINK 动态结构图的时域响应仿真

根据系统的 SIMULINK 动态结构图，可用 linmod ( ) 或 linmod2 ( ) 函数命令对结构图模型提取一个平衡工作点处的线性模型。有了线性模型，不论是指令方式下的时域仿真还是菜单方式下的时域仿真都不会有任何问题。

需要特别注意系统的SIMULINK动态结构图模型.mdl文件在计算机磁盘空间里的存放路径，并注意区别在本书中的插图编号与在MATLAB中的SIMULINK结构图模型.mdl文件名是不同的，但确是指代同一系统。

4.4.5.1 利用SIMULINK动态结构图指令方式下的时域响应仿真实例

这种情况下的系统SIMULINK动态结构图中，输入与输出环节必须是采用模块库浏览器中的“Signals & Systems”中的“In1”与“Out1”模块，因为进行何种仿真是由命令确定的。

【例4-27】 已知直流传动转速单闭环系统的SIMULINK动态结构图如图4-27所示，试绘制出该系统的单位阶跃给定响应与单位斜坡响应曲线。

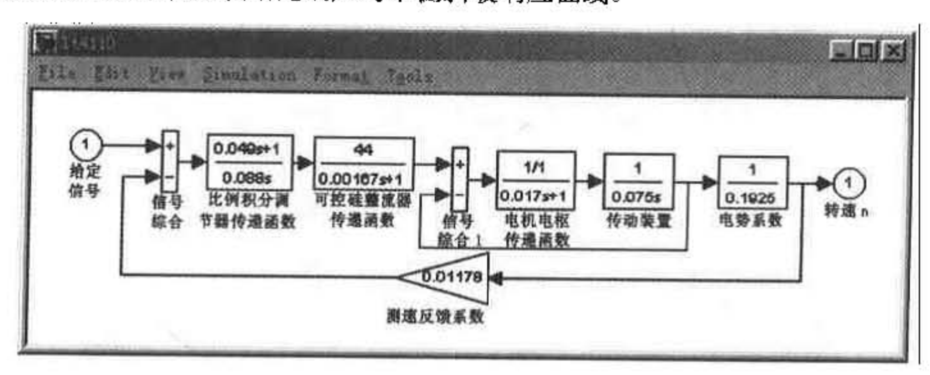


图4-27 转速单闭环系统SIMULINK动态结构图

【解】

根据题目给定的直流转速单闭环调速系统SIMULINK动态结构图（lt4110.mdl），可用linmod()函数命令将lt4110.mdl结构图模型中提取一个在某一平衡工作点处的线性模型。详见3.7节。用函数命令编写MATLAB程序L4408.m如下：

```
% MATLAB PROGRAM L4408.m
%
[a,b,c,d]=linmod('lt4110');[nc,dc]=ss2tf(a,b,c,d);
s1=tf(nc,dc);figure(1);
step(s1);hold on
ddc=[dc,0];figure(2);sys=tf(nc,ddc);t=[0:0.6:60]';
step(sys)
```

程序执行后，即可得到如图4-28所示的单位阶跃给定响应与如图4-29所示的单位斜坡响应曲线。

4.4.5.2 在SIMULINK窗口菜单方式下的时域响应仿真实例

这种情况下的系统SIMULINK动态结构图中，输入与输出环节不能是模块库浏览器中的“Signals & Systems”中的“In1”与“Out1”模块，最好采用进行仿真的信号源模块与观察仿真输出模块。因为菜单方式下的仿真，系统模块必须连接好，设置仿真参数及示波器参数与系统结构图无直接关系。

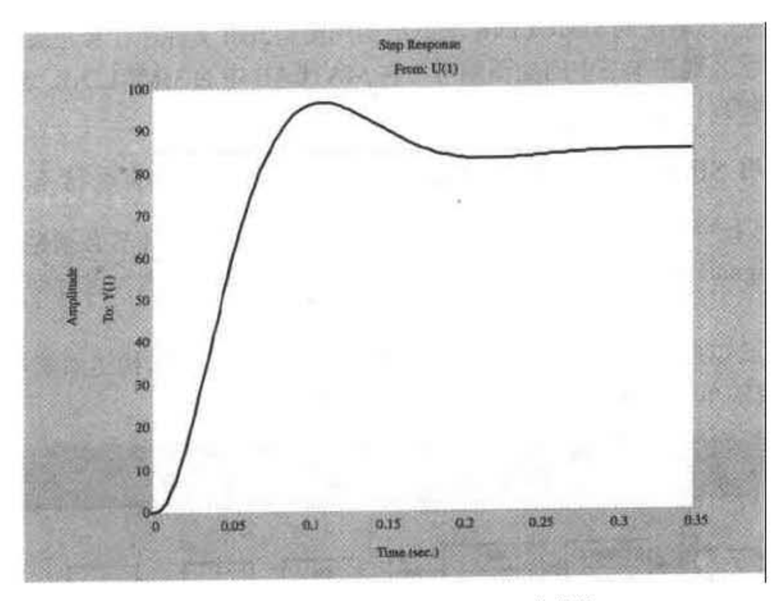


图 4-28　单环系统的阶跃给定响应曲线

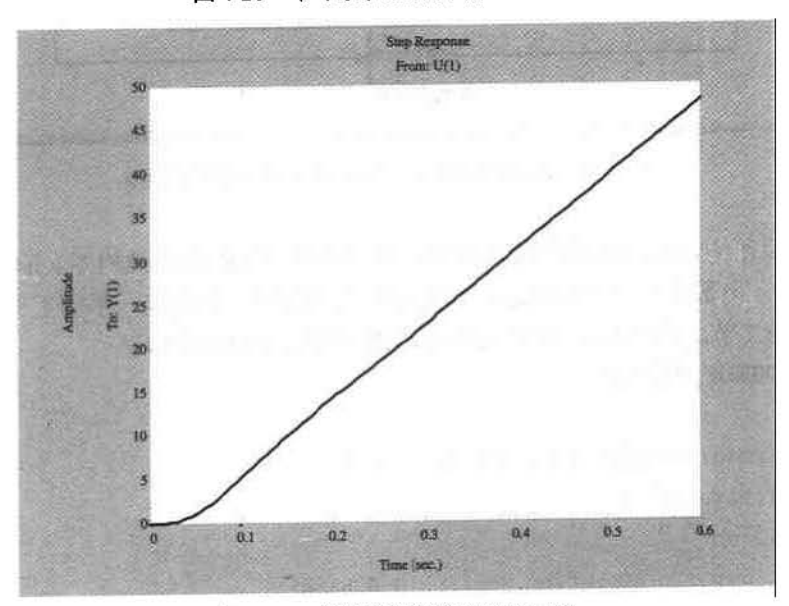


图 4-29　单环系统的斜坡响应曲线

在 SIMULINK 窗口菜单操作方式下进行仿真已经在 3.5 节里做了详尽的介绍，在 SIMULINK 窗口菜单操作方式下进行时域仿真，可以与其他仿真一样操作。

【例 4-28】已知单位负反馈系统如图 4-30 所示，试在 SIMULINK 窗口菜单操方式下进行时域仿真，并做出其单位阶跃响应曲线。

【解】

在 SIMULINK 窗口主菜单里选择【Simulation】,再在其下拉菜单里选择【Start】，即可得到本系统在单位阶跃信号作用下的响应曲线。此响应曲线是在示波器 Scop 里显示出的（参见图 4-31）。

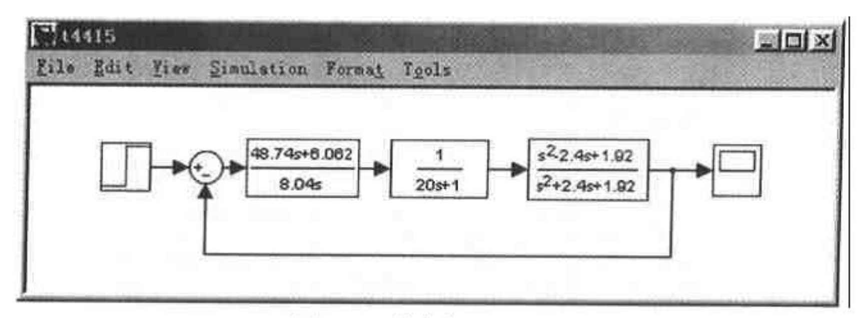


图 4-30　单位负反馈系统

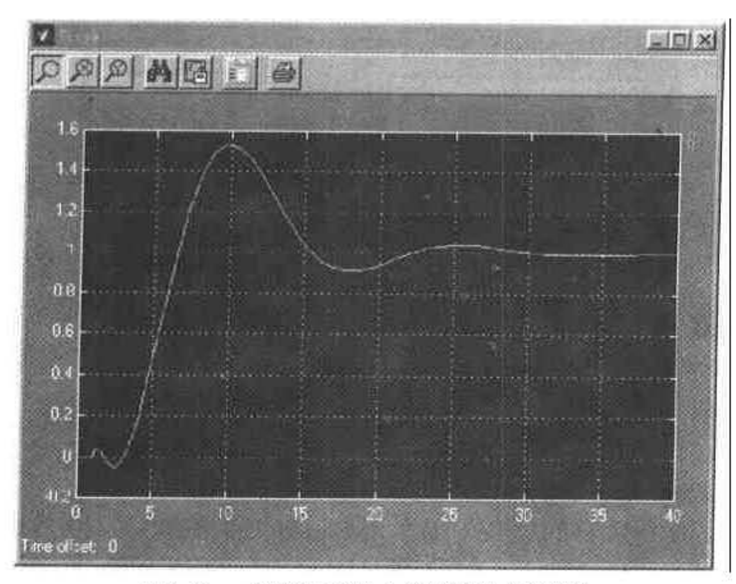


图 4-31　示波器显示单位阶跃给定响应曲线

在示波器 Scop 里，选中菜单按钮“![icon]”，即出现有图 4-32 所示的示波器属性对话框“'Scop' Properties”，可供设置其属性参数。在标签页“General”中，按图 4-32 所示对话框进行设置，然后再按图 4-33 所示的设置仿真参数解算器标签页“Solver”，即可得图 4-31 中示波器 Scop 里显示出的单位阶跃响应曲线。

General | Data history | Tip: try right clicking on axes
Axes
Number of axes: 1　　floating scope
Time range: 40
Tick labels: bottom axis only
Sampling
Decimation　1
OK　Cancel　Help　Apply

图 4-32　设置示波器标签页“General”

图 4-33 设置仿真参数标签页“Solver”

附录：

【MATLAB FUNCTION PROGRAM 2】求系统单位给定阶跃响应性能指标(超调量、余差、阶跃响应的第一（正向）波峰值、第二（正向）波峰值、衰减比、衰减率、阶跃响应的衰减振荡周期与振荡频率)的函数 targ.( )

targ. ( )函数的调用格式为：

[*pm*,*tp*,*ess*,*sigma*,*n*,*pusi*,*T*,*f*]=**targ**(*y*,*t*)

其中，输入参数 *y*,*t* 是系统阶跃响应对应的函数值与其对应的时间。函数返回的是阶跃响应的余差 *ess*、阶跃响应的第一（正向）波峰值 *b*1、阶跃响应的第二（正向）波峰值 *b*2、阶跃响应的超调量 *sgm*、阶跃响应的衰减比 *n*、阶跃响应的衰减率 *pusi*、阶跃响应的衰减振荡周期 *T*、阶跃响应的振荡频率 *f*。

```
function [mp,tp,b1,b2,sigma,n,pusi,T,f]=targ(y,t)
% MATLAB FUNCTION PROGRAM targ.m
%
%Count sigma and tp
[mp,tf]=max(y);
ct=length(t);
mp;
tp=t(tf);
yss=y(ct);
q=1;
m=q-1;
while m<3,
  for a=(tm/100):0.01:tm
   j=[0:a:tm];
```

```
    for i=1:length(j);
        if (y(i+1)-y(i))<0 & (y(i)-y(i-1))>0,
            m=m+1;
            pm(m)=y(i);
             tp(m)=t(i);
        end
      end
  end
end
yss=y(ct);
ess=1-yss;
b1=pm(1)-yss
b2=pm(2)-yss
sigma=100*b1/yss
n=b1/b2
pusi=(b1-b2)/b1
T=(tp(2)-tp(1))
f=1/T
```

【MATLAB FUNCTION PROGRAM 3】求系统单位阶跃给定响应性能指标(超调量、峰值时间、调节时间)的函数 perf. ( )

perf. ( )函数的调用格式为：

[*sigma*,*tp*,*ts*]=**perf**(*key*,*y*,*t*)

其中 *key* 用来选择调节时间的%5 或 2%误差带，当 *key* =1 时表示选择%5 误差带，当 *key* =2 时表示选择 2%误差带。*y*,*t* 是对应系统阶跃响应的函数值与其对应的时间。函数返回的是阶跃响应超调量 *sigma*、峰值时间 *tp*、调节时间 *ts*。

```
function [sigma,tp,ts]=perf(key,y,t)
% MATLAB FUNCTION PROGRAM spxn.m
%
%Count sigma and tp
[mp,tf]=max(y);
cs=length(t);
yss=y(cs);
Sigma=100*(mp-yss)/yss
tp=t(tf)
%Count ts
i=cs+1;
n=0;
while n==0,
   i=i-1;
```

```
            if key==1,
                 if i==1,
                     n=1;
            elseif y(i)>1.05*yss,
                     n=1;
                  end;
             elseif key==2,
                 if i==1,
                    n=1;
                  elseif y(i)>1.02*yss,
                    n=1;
                  end;
             end
end;
t1=t(i);
cs=length(t);
j=cs+1;
n=0;
while n==0,
       j=j-1;
            if key==1,
                 if j==1,
                    n=1;
                 elseif y(j)<0.95*yss,
                    n=1;
                 end;
            elseif key==2,
                 if j==1,
                    n=1;
                 elseif y(j)<0.98*yss,
                    n=1;
                 end;
           end;
end;
t2=t(j);
         if t2<tp
            if t1>t2
               ts=t1
            end
```

```
    elseif t2>tp,
       if t2<t1,
          ts=t2
       else
          ts=t1
       end
    end
```

4.5 控制系统频域分析的 MATLAB 实现

4.5.1 频率特性的有关概念

为了讨论控制系统频域响应的 MATLAB 仿真，有必要复习频率特性的几个基本概念。

1. 频率响应

当正弦函数信号作用于线性系统时，系统稳定后输出的稳态分量仍然是同频率的正弦信号，这种过程叫做系统的频率响应。

2. 频率特性

设有稳定的线性定常系统，在正弦信号作用下，系统输出的稳态分量为同频率的正弦信号，其振幅与输入正弦信号振幅比相对于正弦信号角频率间的关系 $A(\omega)$ 叫做幅频特性；其相位与输入正弦信号的相位之差相对于正弦信号角频率间的关系 $\varphi(\omega)$ 叫做相频特性。系统频率响应与输入正弦信号的复数比叫做系统的频率特性。记作：

$$G(j\omega)=A(\omega)e^{j\varphi(\omega)} \tag{4-37}$$

系统的频率特性与系统传递函数之间有着简单而直接的关系：

$$G(j\omega)=G(s)\big|_{s=j\omega} \tag{4-38}$$

3. 波德图

即对数频率特性曲线。波德图有两条曲线分别是对数幅频特性与对数相频特性。横坐标都是角频率 ω，是按常用对数刻度的即 $\log\omega$，其单位是 rad/s。对数相频特性的纵坐标是 $\varphi(\omega)$，等分刻度，其单位为角度(°)；而对数幅频特性的纵坐标是 $L(\omega)=20\log A(\omega)$，也是等分刻度的，其单位为 dB(即分贝)。

对数频率特性曲线即波德图，在理论教学中特别强调渐近对数频率特性曲线作图对于近似计算的方便作用。但是在计算机仿真里，这种方便的近似计算已经毫无意义了，因为计算机能够做繁琐而精确的计算。

4. 幅相特性

系统的频率特性为：

$$G(j\omega)=A(\omega)e^{j\varphi(\omega)}$$

式中既有振幅信息又有相位信息，所以又叫做系统的幅相特性。幅相特性图形化的形式，即幅相特性曲线。

5．频率特性的极坐标图

频率特性$G(j\omega)$是 ω 的复变函数，在$G(j\omega)$复平面上对于某一 ω 可以用一矢量或其端点（坐标）来表示。当 ω 从0~∞时，$G(j\omega)$ 端点的极坐标轨迹即是频率特性的极坐标图，叫做 Nyquist 曲线（图）。

6．尼柯尔斯图

对于单位负反馈系统其闭环与开环传递函数$G(s)$间的关系为

$$\Phi(s)=\frac{G(s)}{1+G(s)} \tag{4-39}$$

则频率特性之间的关系为：

$$\Phi(j\omega)=\frac{G(j\omega)}{1+G(j\omega)} \tag{4-40}$$

若以模幅式表示$G(j\omega)$有：

$$G(j\omega)=|G|e^{j\angle G} \tag{4-41}$$

则

$$\Phi(j\omega)=M(\omega)e^{j\alpha(\omega)} \tag{4-42}$$

将闭环幅频$M(\omega)$与闭环相频$\alpha(\omega)$合成幅相特性为：

$$M(\omega)e^{j\alpha(\omega)}=\frac{|G|e^{j\angle G}}{1+|G|e^{j\angle G}} \tag{4-43}$$

由此幅相特性可以求得两个函数表达式

$$M=f_1(|G|,\angle G) \tag{4-44}$$

$$\alpha=f_2(|G|,\angle G) \tag{4-45}$$

这两个函数表达式的图形化曲线叫做 Nichols 曲线（图）。

7．频域性能指标

峰值　是幅频特性$A(\omega)$的最大值。

频带　是幅频特性$A(\omega)$的数值衰减到0.707 $A(0)$时对应的频率。

相频宽　是相频特性$\varphi(\omega)$等于$-\pi/2$时对应的频率。

8．稳定裕度

相角稳定裕度　是指系统开环幅相特性曲线 $G(j\omega)$上模值等于 1 的矢量与负实轴的夹角：

$$\gamma=\varphi(\omega_c)-(-\pi) \tag{4-46}$$

模值稳定裕度　是指系统开环幅相特性曲线$G(j\omega)$与负实轴交点（ω_{cg}）模值$G(\omega_{cg})$的倒数：

$$h = \frac{1}{\left|G(\omega_{cg})\right|} \tag{4-47}$$

或者是其交点模值倒数的分贝值：

$$L_h = 20\log h \tag{4-48}$$

9．剪切频率

是指系统开环对数振幅频率特性曲线 $20\log|G|$ 与横坐标轴（ω）的交点，常用 ω_c（或 ω_{cp}）来标识。

10．-π 穿越频率

是指系统开环对数相频特性曲线 $\varphi(\omega)$ 与-π 线交点所对应的角频率（即系统开环幅相特性曲线 $G(j\omega)$ 与负实横轴的交点所对应的角频率），常用 ω_g（或 ω_{cg}）来标识。

11．三频段概念

系统开环对数振幅特性曲线 $20\log|G|$ 按横坐标大致分为三个频段。在开环对数振幅特性曲线 $20\log|G|$ 上第一个转折频率以左的区段，是由积分环节与开环增益所确定的，此为低频段；在 $20\log|G|$ 上第一个转折频率以右的剪切频率 ω_c 附近的区段叫做中频段，这段特性集中反映了闭环系统动态响应的稳定性与快速性。在中频段以右（$\omega > 10\omega_c$）的区段叫做高频段，此部分是由系统中时间常数很小频带很高的元件所决定的，系统在高频段的分贝值，反映了系统对输入高频干扰信号的抑止能力。

12．乃奎斯特（Nyquist）稳定判据

系统开环幅相特性 $G(j\omega)$ 如果满足以下关系式：

$$\Delta \angle_{\omega:\,0\to\infty} \left[1+G(j\omega)\right] = p\pi \tag{4-49}$$

则系统闭环稳定。式中 p 为系统开环特征方程的不稳定根个数。

13．对数频率稳定判据

当 p=0 时，在开环对数幅相特性曲线 $20\log|G|>0$ 的范围内，相频特性曲线 $\varphi(\omega)$ 对-π 线的正穿（由下向上）次数与负穿（由上向下）次数相等，则系统闭环稳定；当 $p \neq 0$ 时，在开环对数幅相特性曲线 $20\log|G|>0$ 的范围内，若相频特性曲线 $\varphi(\omega)$ 对-π 线的正穿次数与负穿次数之差为 $p/2$，则系统闭环稳定。

4.5.2 系统频域分析的内容与 MATLAB 仿真的主要问题

频域分析法是自动控制中应用又一种数学工具——频率特性来研究系统控制过程性能：稳定性、快速性及稳态精度的方法。这种方法不必直接求解系统的微分方程，而是间接地运用系统的开环频率特性曲线，分析闭环系统的响应，因此它是一种图解的方法。

频域分析里主要用到前述三种曲线（或叫图）：Bode 图、Nyquist 曲线图、Nichols 曲线图。这三种曲线（或图）就是频率分析的三种工具。

Bode 图可以用于分析系统的相角稳定裕度、幅值稳定裕度、剪切频率、-π 穿越频率、带宽、扰动抑止及其稳定性等，所以 Bode 图在频域分析里占有重要的地位。

Nyquist 与 Nichols 曲线图也可以用来分析系统的相角稳定裕度、幅值稳定裕度及其稳定性等，在频域分析里也很有用。

对于频域分析，本书论述的 MATLAB 仿真就是用先进的计算机仿真技术来精确绘制三种曲线，并计算系统的频域性能指标：剪切频率ω_c、$-\pi$穿越频率ω_g、相角稳定裕度γ、幅值稳定裕度L_h，以便研究系统控制过程的稳定性、快速性及稳态精度等性能。

4.5.3 频域分析 MATLAB 实现的方法

频域分析在经典控制里占有重要的地位。传统的频域分析方法是计算数据，绘制 Bode 图，并求出频域性能指标。这种老办法要耗费大量的时间、精力，而且计算数据的精确度还不一定能够得到保证。借助于计算机及其软件，特别是利用 MATLAB 软件，运行它提供频率分析的函数，能够方便、简单、快捷地绘制三种曲线： Bode 图、 Nyquist 曲线图、Nichols 曲线图，并计算出频域性能指标，还可以借助于这些曲线对系统进行分析。

应用 MATLAB 提供的上述频域仿真函数与其他函数命令、语句，编制成 MATLAB 程序，这种 MATLAB 的指令方式下进行频域仿真同样是最常用的实现方法。

4.5.3.1 频域响应 MATLAB 仿真的函数指令格式

1．求连续系统 Bode 图的函数 bode ( )

函数命令调用格式：

[*mag*,*phase*,*w*] = **bode** (*sys*)

bode (*sys*)

bode (*sys*, *w*)

bode (*sys*1,*sys*2,…,*sysn*)

bode (*sys*1,*sys*2,…,*sysn*,*w*)

bode(*sys*1,'*PlotStyle*1',…*sysn*,'*PlotStylen*')

函数命令使用说明：

bode ( ) 函数用来计算并显示绘制的系统 Bode 图，可用于 SISO 或者 MIMO 的连续时间系统或者离散时间系统。当函数命令为无等式左边输出变量的格式时，函数在当前图形窗口中直接绘制出系统的 Bode 图。LTI 对象 sys 可以是由函数 tf ( )、zpk ( )、ss ( ) 中任何一个函数建立的系统模型。计算离散时间系统阶跃响应的函数为 dbode ( )。

bode (*sys*, *w*) 函数用于显示绘制的系统 Bode 图，函数中输入参数ω用来定义绘制 Bode 图时的频率范围或者频率点。若要定义频率范围，ω必须为［wmin,wmax］格式；如果定义频率点，则ω必须为由需要频率点频率构成的向量。

bode (*sys*1,*sys*2,…,*sysn*,*w*) 函数可同时在一个窗口绘制多个 LTI 对象的 Bode 图。这些系统必须有同样数量的输入与输出，但可以同时含有连续与离散时间系统。显然，当多个系统需要比较其频域性能指标时，这个函数命令是用得着的。

bode (*sys*1,'*PlotStyle*1',…*sysn*,'*PlotStylen*') 函数可在绘制 Bode 图时设置绘制曲线的属性，其 *PlotStyle* 应是 MATLAB 标准函数命令 plot（） 支持的各种属性标识符字符串。

[**mag**,**phase**,*w*] = bode (*sys*)或者[**mag**,**phase**,*w*] = bode (*sys*,*w*) 函数为带有输出变量引

用的函数，可计算系统 Bode 图的输出数据，而不绘制出曲线。输出变量 *mag* 是系统 Bode 图的振幅值；输出变量 *phase* 为 Bode 图的相位值；输出变量 w 即 ω 是系统 Bode 图的频率点。其中，***mag*** 与 ***phase*** 都是 3 维向量，其大小为：(number of outputs)*(number of inputs)*(length of w)。对于 SISO 系统，有：

$$\begin{cases}\omega_k = W(k)\\ \boldsymbol{mag}(1,1,k) = |G(j\omega)|\\ \boldsymbol{phase}(1,1,k) = \angle G(j\omega)\end{cases} \tag{4-50}$$

对于 MIMO 系统，第 i 个输入到第 j 个输出之间的传递函数 G_{ij}，有：

$$\begin{cases}\boldsymbol{mag}(i,j,k) = |G_{ij}(j\omega)|\\ \boldsymbol{phase}(i,j,k) = \angle G_{ij}(j\omega)\end{cases} \tag{4-51}$$

以上关系式中，角频率 ω 以弧度/秒为单位，相位 ***phase*** 以度（°）为单位。必须特别是指出，按式（4-50）或式（4-51）计算的幅值 ***mag*** 不是分贝值，但是可以用以下公式转换成分贝值：

$$\boldsymbol{magdb} = 20 * \log(\boldsymbol{mag}) \tag{4-52}$$

【例 4-29】由集成运算放大器构成的积分器如图 4-34 所示：

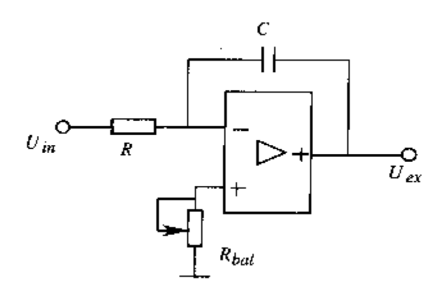


图 4-34 集成运算放大器构成的积分器

求当 $R = 10\,\text{k}\Omega, C = 0.0025\,\mu\text{F}$ 时的 Bode 图。

【解】

由集成运算放大器构成积分器的传递函数为

$$G(s) = \frac{\boldsymbol{num}(s)}{\boldsymbol{den}(s)} = \frac{1}{RCs} = \frac{1}{Ts}$$

根据题目要求，用函数命令给出 MATLAB 程序段如下：

```
R=10*1000;C=0.0025/1000000;
T=R*C; num=[1];den=[T 0];
sys=tf(num,dden);bode(sys)
```

运行该程序段可得积分器的 Bode 图，如图 4-35 所示。从图中的相频曲线可以看出积分器的相频角为-90° 即-π/2。

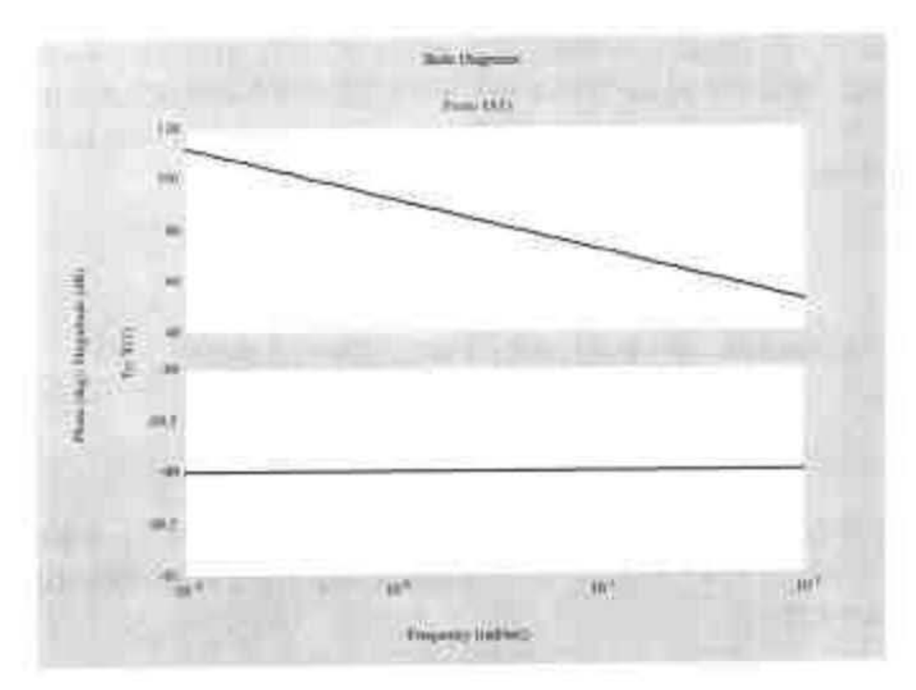


图 4-35 积分器的 Bode 图

很容易验算，当 $\omega=1$ rad/s 时，对数幅频特性曲线的分贝值为：

```
1/T=1/0.000025;
20*log10(1/T)
ans =
   92.0412  dB
```

即 L=20*log(1/T)=92.0412 dB 与图 4-35 一致。

2. 求离散系系统 bode 图的函数 dbode（）

函数命令调用格式：

[*mag*, *phase*, *w*] = **dbode** (*a*, *b*, *c*, *d*,*Ts*)

[*mag*, *phase*, *w*] = **dbode** (*a*, *b*, *c*, *d*, *Ts*, *iu*)

[*mag*, *phase*, *w*] = **dbode** (*a*, *b*, *c*, *d*, *Ts*, *iu*, ω)

[*mag*, *phase*, *w*] = **dbode** (***num***, ***den***,*Ts*)

[*mag*, *phase*, *w*] = **dbode** (***num***, ***den***,*Ts*, ω)

函数命令使用说明：

dbode(num,den,Ts) 函数可计算并绘制出 LTI 离散时间系统的幅频与相频响应曲线（即 Bode 图）。当缺省输出变量引用函数时，dbode 函数可在当前图形窗口中直接绘制出 LTI 系统的 Bode 图。(num,den)分别是脉冲传递函数 $G(z)=\dfrac{\boldsymbol{num}(z)}{\boldsymbol{den}(z)}$ 的分子与分母，***num***(*z*) 为分子 *z* 的多项式，***den***(*z*) 为分母 *z* 的多项式。***num***(*z*) 与 ***den***(*z*) 两者均按 *z* 的递减幂次排列。*Ts* 是内部采样时间。

dbode (*a*, *b*, *c*, *d*,*Ts*) 可以绘制系统的一组 Bode 曲线图，每一条曲线对应于离散状态空间系统

$$\begin{cases} x(k+1) = a\ x(k) + b\ u(k) \\ y(k) = c\ x(k) + d\ u(k) \end{cases} \tag{4-53}$$

的每个输入，其角频率范围是函数自动选取的。频率点在 0 到 $\frac{\pi}{Ts}$ 之间选取。而在响应快速变化的位置系统会自动选用更多的采样点。*Ts* 是内部采样时间。

dbode (a, b, c, d, *Ts*,*iu*)可绘制出从第 *iu* 个输入到所有输出的系统 Bode 图。

dbode (*a*, *b*, *c*, *d*, *Ts*, *iu*, *ω*) 或 dbode (***num***, ***den***, *Ts*, *ω*) 可利用用户指定的角频率范围 *ω* 来绘制 Bode 图。

当带有输出变量引用函数时，可得到系统 bode 图与之相应的数据，而不直接绘制出 Bode 图，幅值及相位可根据以下公式计算：

$$\begin{cases} G(z) = c(zI - a)^{-1} b + d \\ mag(\omega) = |G(j\omega)| \\ phase(\omega) = \angle G(j\omega) \end{cases} \tag{4-54}$$

式中相位 ***phase*** 以度(°)为单位，角频率 *ω* 以弧度/秒（rad/s）为单位。同样必须注意，按式（4-54）计算的幅值 ***mag*** 不是分贝值，但是可以用以下公式换算成分贝值：

$$magdb = 20 * \log(mag)$$

【例 4-30】已知二阶离散系统开环的脉冲传递函数为：

$$G(z) = \frac{0.7z + 0.06}{z^2 - 0.5z + 0.43}$$

求离散系统当 *Ts*=0.1 s 时的 Bode 图。

【解】

根据题目要求，用函数命令给出如下程序段：

```
num=[0.7 0.06];den=[1 -0.5 0.43];
dbode(num,den,0.1)
```

运行该程序可得二阶离散系统的 Bode 图，如图 4-36 所示。

3．计算（绘制）系统 Nyquist 曲线的函数 nyquist ( )

函数命令调用格式：

[*re*,*im*,*w*] = **nyquist**(*sys*)

nyquist(*sys*)

nyquist(*sys*, *w*)

nyquist (sys1,*sys*2,…,*sysn*)

nyquist(*sys*1,sys2,…,*sysn*,*w*)

nyquist(*sys*1,'*PlotStyle*1',…*sysN*,'*PlotStyleN*')

函数命令使用说明：

nyquist ( ) 函数用来计算并显示绘制的系统奈奎斯特曲线，可用于 SISO 或者 MIMO 的连续时间系统或者离散时间系统。当函数命令为无等式左边输出变量的格式时，函数在当前图形窗口中直接绘制出系统的 Nyquist 曲线。LTI 对象 *sys* 可以是由函数 tf ( )、zpk

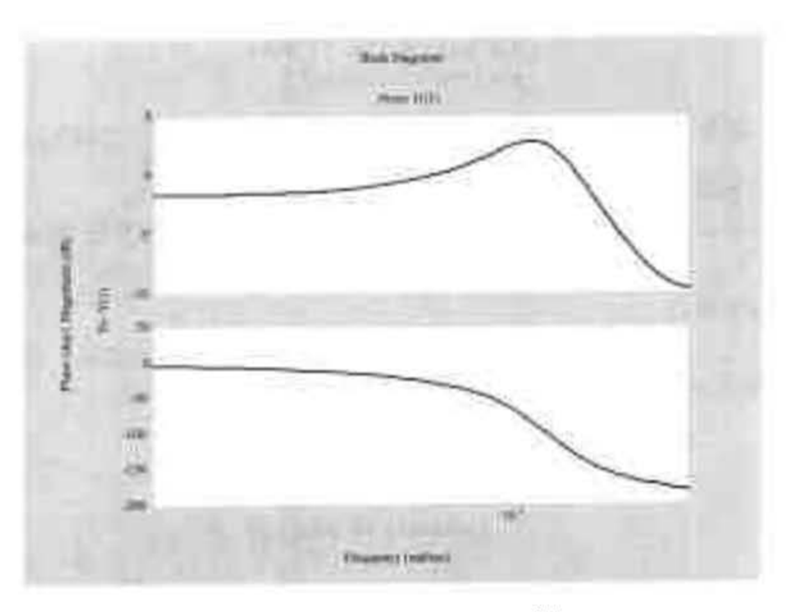


图 4-36　二阶离散系统的 Bode 图

( )、ss ( ) 中任何一个函数建立的开环系统模型。计算并显示绘制离散系统 Nyquist 曲线的函数为 dnyquist ( )。

nyquist(*sys*, *w*) 函数用于显示绘制的系统 Nyquist 曲线，函数中输入参数 ω 用来定义绘制 Nyquist 曲线时的频率范围或者频率点。若要定义频率范围，ω 必须是 [wmin,wmax] 格式；如果定义频率点，则 ω 必须为由需要频率点频率构成的向量。

nyquist (*sys*1,*sys*2,…,*sysn*,*w*) 函数可同时在一个窗口绘制多个 LTI 对象的 Nyquist 曲线。这些系统必须有相同的输入与输出个数，但可以同时含有连续与离散时间系统。显然，当多个系统需要比较其 Nyquist 曲线时，这个函数命令是有用的。

nyquist(*sys*1,'*PlotStyle*1',…*sysn*,'*PlotStylen*') 函数可在绘制 Nyquist 曲线时设置绘制曲线的属性，其 PlotStyle 应是 MATLAB 标准函数命令 plot ( ) 支持的各种属性标识符字符串。

[***re***,***im***,*w*] = nyquist(*sys*)或者[***re***,***im***,*w*] = nyquist(*sys*,*w*)函数为带输出变量引用的函数，可计算系统在频率 ω 处的频率响应输出数据，而不绘制出曲线。其中，输出变量 re 为频率响应的实部，***im*** 为频率响应的虚部，ω 是频率点。***re*** 与 ***im*** 都是 3 维向量，其大小为：(number of outputs)*(number of inputs)*(length of w)。对于 SISO 系统，有：

$$\begin{cases}\omega_k = W(k)\\ \boldsymbol{re}(1,1,k)=\boldsymbol{re}(G(j\omega))\\ \boldsymbol{im}(1,1,k)=\boldsymbol{im}(G(j\omega))\end{cases} \tag{4-55}$$

对于 MIMO 系统第 i 个输入与第 j 个输出之间的传递函数 G_{ij}，有：

$$\begin{cases}\boldsymbol{re}(i,j,k)=\boldsymbol{re}(G_{ij}(j\omega))\\ \boldsymbol{im}(i,j,k)=\boldsymbol{im}(G_{ij}(j\omega))\end{cases} \tag{4-56}$$

nyquist ( ) 函数绘制的开环系统 Nyquist 曲线可以用来确定系统闭环的稳定性。当

已绘制出开环系统传递函数$G(s)$的 Nyquist 曲线时，如果 Nyquist 曲线按逆时针方向包围（-1+j0）点 p 次（p 为系统开环特征方程中不稳定根的个数），则闭环系统

$$\Phi(s)=\frac{G(s)}{1+G(s)} \tag{4-57}$$

是稳定的。这就是 Nyquist 稳定判据。应用 Nyquist 稳定判据必须先绘制出 Nyquist 曲线，由此可见 nyquist（）函数与 Nyquist 稳定判据是密不可分的。

【例 4-31】已知直流单闭环系统的 SIMULINK 动态结构如图 4-37 所示。图中转速闭环已经断开。已知 K_p=2；$K_S=22$；$T_s=0.00167$(s)；$R=1\,\Omega$；$T_l=0.017$ (s)；$C_e=0.1925$ V/(r/min)；$T_m=0.075$ (s)；$\alpha=0.01178$ V/(r/min)；试绘制出该系统的 Nyquist 曲线，并用 Nyquist 稳定判据对闭环系统判稳。

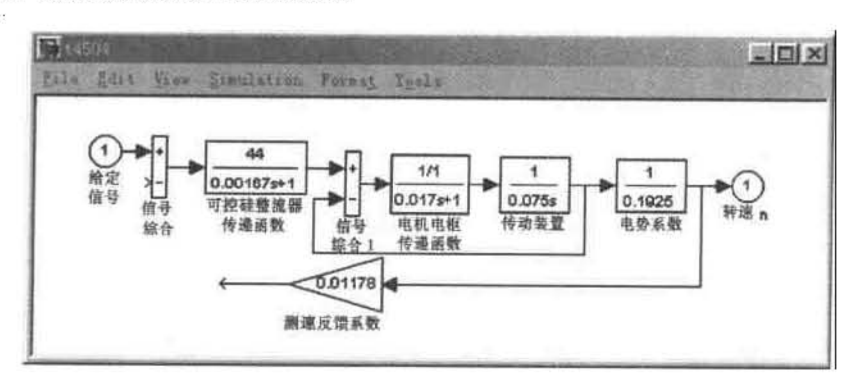


图 4-37　转速单闭环系统前向通道的 SIMULINK 动态结构图

【解】

根据题目已知条件，直流传动转速单闭环系统的 SIMULINK 动态结构图，用函数命令编写 MATLAB 程序 L4503.m。

```
%MATLAB PROGRAM L4503a.m
%
n1=[1];d1=[0.017 1];s1=tf(n1,d1);
n2=[1];d2=[0.075 0];s2=tf(n2,d2);
sys1=feedback(s1*s2,1);
n3=[0 44];d3=[0.00167 1];s3=tf(n3,d3);
n4=[0 1];d4=[0 0.1925];s4=tf(n4,d4);
n5=[0 0.01178];d5=[0 1];s5=tf(n5,d5);
G=sys1*s3*s4*s5;nyquist(G)
```

程序执行后，即可得到如图 4-38 所示的 Nyquist 曲线。

为了应用 Nyquist 稳定判据对闭环系统判稳，必须知道 $G(s)$不稳定根个数 p 是否为 0。为此，求出系统开环的传递函数，并求其特征方程的根。执行以下 L4503b.m 程序求开环的传递函数。

```
%MATLAB PROGRAM L4503b.m
%
```

```
n1=[1];d1=[0.017 1];s1=tf(n1,d1);
n2=[1];d2=[0.075 0];s2=tf(n2,d2);
sys1=feedback(s1*s2,1);
n3=[0 44];d4=[0.00167 1];s4=tf(n4,d4);
n4=[0 1];d5=[0 0.1925];s5=tf(n5,d5);
n5=[0 0.01178];d6=[0 1];s6=tf(n6,d6);
G=sys1*s3*s4*s5
Transfer function:
                      2.156 s + 44
---------------------------------------------------------------
3.607e-008 s^4 + 2.372e-005 s^3 + 0.001299 s^2 + 0.01694 s
```

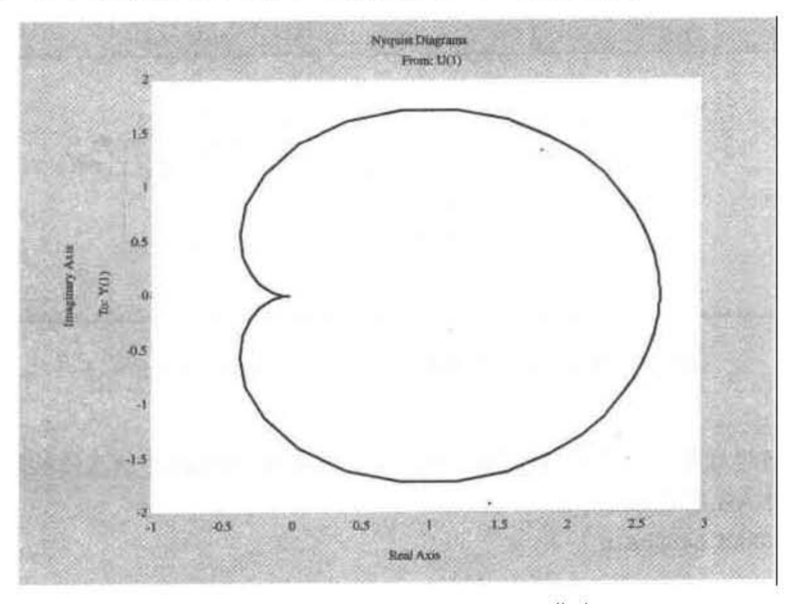


图 4-38　系统前响通道的 Nyquist 曲线

再执行以下指令求其特征方程的根：

```
p=[3.0607e-008 2.372e-005 0.001299 0.01694];
roots(p)
ans =
 -716.8586
 -37.5859
  -20.5416
```

由运算数据可知，特征方程的根全为稳定根，即 $p=0$。

图 4-38 中 nyquist 曲线没有包围且远离（ −1+j0）点。另外，$G(s)$不稳定根个数 p=0，所以由 $G(s)$构成的闭环系统是稳定的。

4. 计算（绘制）离散系统 Nyquist 曲线的函数 dnyquist（）

函数命令调用格式：

[*re*,*im*,*w*] = dnyquist (*a*, *b*, *c*, *d* ,*Ts*)
[*re*,*im*,*w*] = dnyquist (*a*, *b*, *c*, *d*, *Ts*, *iu*)
[*re*,*im*,*w*] = dnyquist (*a*, *b*, *c*, *d*, *Ts*, *iu*, *ω*)
[*re*,*im*,*w*] = dnyquist (*num*, *den*, *Ts*)
[*re*,*im*,*w*] = dnyquist (*num*, *den*, *Ts*, *ω*)

函数命令使用说明：

dnyquist 函数可计算并绘制出离散时间 LTI 系统的 Nyquist 曲线。当缺省输出变量引用函数时，dnyquist（） 函数可在当前图形窗口中绘制出 LTI 系统的 Nyquist 曲线。

dnyquist (***num***,***den***,***Ts***) 可绘制出脉冲传递函数 $G(z)=\dfrac{\boldsymbol{num}(z)}{\boldsymbol{den}(z)}$ 对应系统的 Nyquist 曲线。*Ts* 是内部取样时间。输入参数 (***num***,***den***) 是系统开环脉冲传递函数 $G(z)=\dfrac{\boldsymbol{num}(z)}{\boldsymbol{den}(z)}$ 的分子与分母，***num***(*z*) 为分子 *z* 的多项式，***den***(*z*) 为分母 *z* 的多项式。***num***(*z*) 与 ***den*** (*z*)两者都按 *z* 的递减幂次排列。

dnyquist (***a***, ***b***, ***c***, ***d***, *Ts*) 可以绘制系统的一组 Nyquist 曲线，每条曲线相应于离散状态空间系统

$$\begin{cases} \boldsymbol{x}(k+1)=a\,x(k)+b\,\boldsymbol{u}(k) \\ \boldsymbol{y}(k)=c\,x(k)+d\,\boldsymbol{u}(k) \end{cases}$$

的输入/输出组合对，其角频率范围是函数自动选取的，频率点在 0 到 $\dfrac{\pi}{Ts}$ 弧度之间选取。而在响应快速变化的位置系统会自动选用更多的采样点。*Ts* 也是内部取样时间。

dnyquist (***a***, ***b***, ***c***, ***d***, ***Ts***, ***iu***)可绘制出从第 *iu* 个输入到所有输出的 Nyquist 曲线。

dnyquist (***a***, ***b***, ***c***, ***d***, *Ts*, *iu*, *ω*) 或 dnyquist (***num***, ***den***, *Ts*, *ω*) 可利用用户设定的角频率向量 *ω* 来绘制系统的 Nyquist 曲线。

当带输出变量引用函数时，可得到与系统 Nyquist 曲线相应的输出数据，而不直接绘制出 Nyquist 曲线。

dnyquist（） 函数也可以用来确定离散系统的稳定性。如果已经绘制出离散系统开环的 Nyquist 曲线，并且 Nyquist 曲线按逆时针方向包围（-1+j0）点 *p* 次（*p* 为不稳定开环极点的个数），则闭环系统

$$\Phi(z)=\frac{G(z)}{1+G(z)} \tag{4-58}$$

是稳定的。

【例 4-32】已知二阶离散系统开环的脉冲传递函数为：

$$G(z)=\frac{0.7z+0.06}{z^2-0.5z+0.43}$$

求离散系统当 *Ts*=0.1 s 时的 Nyquist 曲线，并用 Nyquist 稳定判据对闭环系统判稳。

【解】

根据题目要求，给出以下调用函数命令的程序段：

```
num=[0.7 0.06];
den=[1 -0.5 0.43];
dnyquist(num,den,0.1)
```

程序段执行后，即可得到二阶离散系统的 nyquist 曲线，如图 4-39 所示。

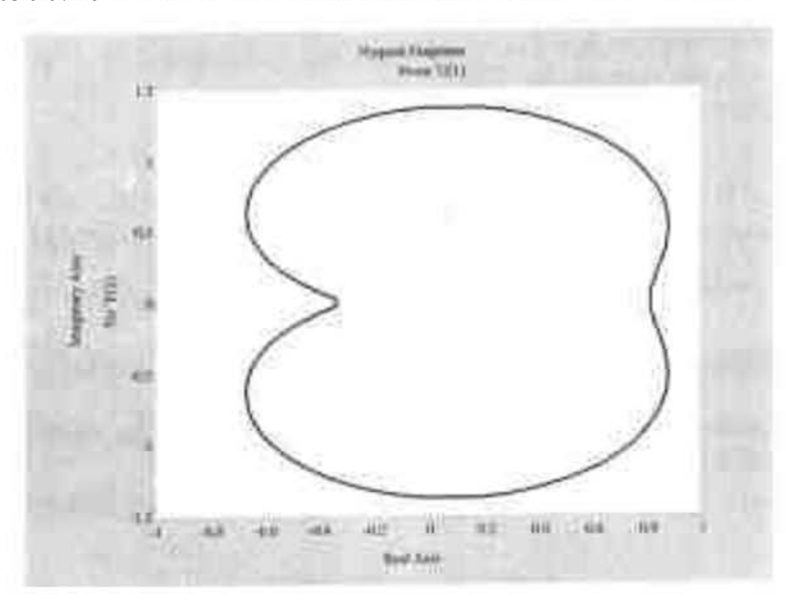


图 4-39　二阶离散系统的 nyquist 曲线

为了应用 Nyquist 稳定判据对闭环系统判稳，必须知道 $G(z)$ 不稳定根个数 p 是否为 0。为此，需要判断离散闭环特征方程的根是否全部位于 z 平面的单位圆内或者将 z 平面映射到可以利用代数稳定判据的平面（例如 w 平面），然后求其特征方程的根是否为稳定根。由离散系统开环的脉冲传递函数

$$G(z)=\frac{0.7z+0.06}{z^2-0.5z+0.43}$$

可以得闭环特征方程

$$D(z)=z^2+0.2z+0.49=0$$

求特征方程的根:

```
p=[1 0.2 0.49];
roots(p)
```

指令运行后可得:

```
ans =
   -0.1000 + 0.6928i
   -0.1000 - 0.6928i
```

特征方程的根都位于 z 平面的单位圆内，全为稳定根。

也可以作 W 代换:

$$z=\frac{w+1}{w-1} \tag{4-59}$$

可以得到 w 平面里的特征方程:

$$D(\mathrm{w}) = 1.69w^2 + 1.02w + 1.29 = 0$$

求 w 平面里特征方程的根：

```
p=[1.69 1.02 1.29];
roots(p)
```

指令运行后可得：

```
ans =
  -0.3018 + 0.8199i
  -0.3018 - 0.8199i
```

特征方程根的实部全为负，都位于 w 平面的左侧，特征方程的根也全为稳定根。

图 4-39 中 nyquist 曲线没有包围（−1+j0）点，且 $G(z)$ 不稳定根个数 p=0，所以由 $G(z)$ 构成的闭环系统是稳定的。

5．求连续系统 Nichols 曲线的函数 nichols ( )

函数命令调用格式：

[***mag*,*phase***,*w*] = **nichols**(*sys*)

nichols(*sys*)

nichols(*sys*, *w*)

nichols(*sys*1,*sys*2,…,*sysN*)

nichols(*sys*1,*sys*2,…,*sysN*,*w*)

nichols(*sys*1,'*PlotStyle*1',…*sysN*,'*PlotStyleN*')

函数命令使用说明：

nichols () 函数用来计算并显示绘制的 LTI 系统 Nichols 频率响应曲线，可用于 SISO 或者 MIMO 的连续时间系统或者离散时间系统。当系统为 MIMO 时，该函数绘制出一组 Nichols 曲线，每个输入输出通道对应一组曲线，绘制曲线时的频率范围将依据系统的零极点来决定。

当函数命令为无等式左边输出变量的格式时，函数在当前图形窗口中直接绘制出系统的 Nichols 曲线。LTI 对象 *sys* 可以是由函数 tf ( )、zpk ( )、ss ( ) 中任何一个建立的系统模型。计算并显示绘制离散系统 Nichols 曲线的函数为 dnichols ( )。

nichols(*sys*, *w*) 函数用于显示绘制的系统 Nichols 曲线，函数中输入参数 *w* 即 $\boldsymbol{\omega}$ 用来定义绘制 Nichols 曲线时的频率范围或者频率点。若要定义频率范围，$\boldsymbol{\omega}$ 必须是［wmin,wmax］格式；如果定义频率点，则 $\boldsymbol{\omega}$ 必须为由需要频率点频率构成的向量。

nichols(*sys*1,*sys*2,…,*sysn*,*w*) 函数可同时在一个窗口绘制多个 LTI 对象的 Nichols 曲线。这些系统必须有相同的输入与输出个数，但可以同时含有连续与离散时间系统。显然，当多个系统需要比较其 Nichols 曲线时，这个函数命令是有用的。

nichols(*sys*1,'*PlotStyle*1',…*sysn*,'*PlotStylen*') 函数可在绘制 Nichols 曲线时设置绘制曲线的属性，其 *PlotStyle* 应是 MATLAB 标准函数命 plot（）支持的各种属性标识符字符串。

[*mag*,*phase*,*w*] = nichols(*sys*)或者 [*mag*,*phase*,*w*] = nichols(*sys*,*w*)函数为带输出变量引用的函数，可计算系统的频率响应 Nichols 曲线输出数据，而不绘制出曲线。其中，输出

变量 ***mag*** 为频率响应 Nichols 曲线的幅度值，***phase*** 为 Nichols 曲线的相位值，ω 是 Nichols 曲线的频率点。***mag*** 与 ***phase*** 都是 3 维向量，其大小为：(number of outputs)*(number of inputs)*(length of w)。

【例 4-33】设一高阶系统开环的传递函数为：

$$G(s)=\frac{0.0001s^3+0.0218s^2+1.0436s+9.3599}{0.0006s^3+0.0268s^2+0.06365s+6.2711}$$

试绘制出该系统的 Nichols 曲线。

【解】

根据题目要求，给出以下调用函数命令的程序段：

```
num=[0.0001 0.0218 1.0436 9.3599];
den=[0.0006 0.0268 0.6365 6.2711];
sys=tf(num,den);
ngrid('new');
nichols(sys)
```

运行该程序段可得高阶连续系统的 Nichols 图，如图 4-40 所示。

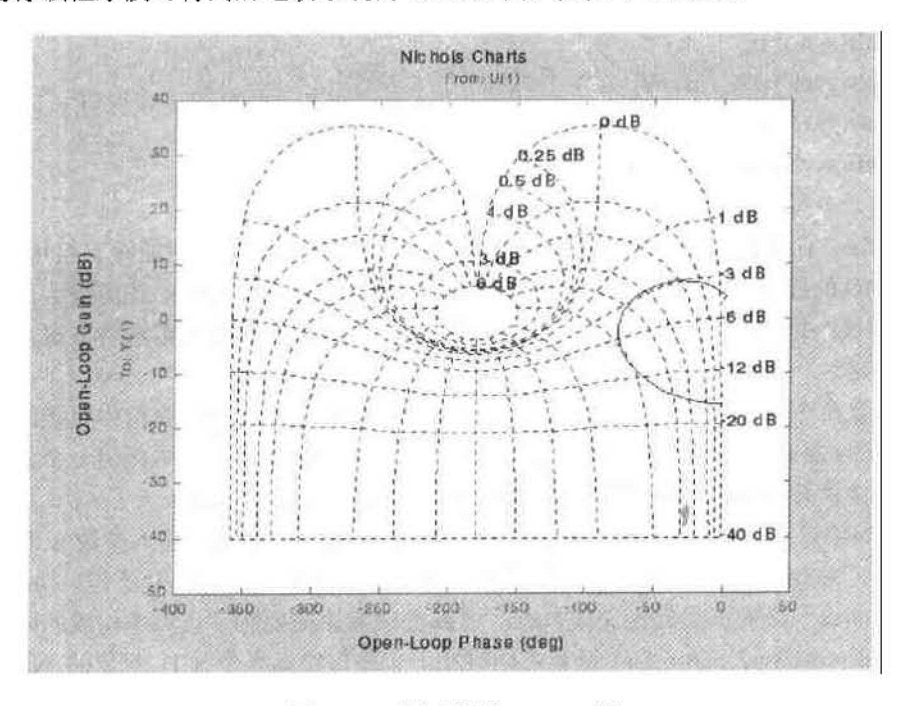


图 4-40　高阶系统的 Nichols 图

6．求系统模值裕度与相位裕度的函数 margin（）

函数命令调用格式：

[*Gm*,*Pm*,*Wcp*,*Wcg*]=**margin** (*sys*)

[*Gm*,*Pm*,*Wcp*,*Wcg*]=**margin** (***mag***,***phase***,ω)

margin (*sys*)

margin (*mag,phase*, *ω*)

函数命令使用说明：

margin（） 函数可以从频率响应数据中计算出模值稳定裕度、相位稳定裕度及其对应的角频率。输入参量 *sys* 一般是用系统的开环传递函数描述的系统模型，对于开环 SISO 系统，既可以是连续时间系统也可是离散时间系统。性能指标模值裕度、相位裕度及其相应的角频率对于控制系统分析、计算及仿真是非常有用的，因为从这些数据即可确定系统闭环时的相对稳定性及其稳定裕量。当不带输出变量引用函数时，margin ( ) 函数可在当前图形窗口有中绘制出带有稳定裕度的 Bode 图。

margin (*mag,phase*, *ω*) 函数可以在当前窗口中绘制出带有系统的模值裕度与相位裕度 Bode 图，其中，*mag,phase* 及 *ω* 分别为由 bode 或 dbode 求出的模值裕度、相位裕度及其对应的角频率。[*Gm,Pm,Wcp,Wcg*]=margin（*mag,phase*, *ω* ）函数是 margin (*mag,phase*,*ω*) 函数带输出变量的引用形式，该函数不绘制 Bode 图，输出变量返回的 Gm 是系统模值裕度及其对应的角频率 *Wcp*，*Pm* 是相位裕度及其对应的角频率 *Wcg*。

[*Gm,Pm,Wcp,Wcg*]=margin (*sys*) 函数命令可计算出系统的频率性能指标，返回的 *Gm* 是系统模值裕度(不是分贝值)及其对应的角频率 *Wcp*，*Pm* 是相位裕度及其对应的角频率 *Wcg*，函数不绘制 Bode 图。

对于离散时间系统，可以先用函数 dbode（）计算频率响应，然后再调用 margin（）函数，即用 margin (*mag,phase*,*ω*)函数命令，再求系统性能指标或绘制 Bode 图。

【例 4-34】已知一单位反馈系统前向通道的传递函数为：

$$G(s)=\frac{2s^4+8s^3+12s^2+8s+2}{s^6+5s^5+10s^4+10s^3+5s^2+s}$$

试绘制出 Bode 图并计算系统的频域性能指标。

【解】

根据题目要求，调用函数命令 margin ( ) 的 MATLAB 程序 L4506.m 如下：

```
%MATLAB PROGRAM L4506.m
%
num=[0 0 2 8 12 8 2];den=[1 5 10 10 5 1 0];
sys=tf(num,den);
[mag,phase,w]=bode(sys);
[gm,pm,wcp,wcg]=margin(mag,phase,w)
margin(mag,phase,w)
```

运行该程序可得系统的 Bode 图，如图 4-41 所示，系统的性能指标如下：

gm = 297.0483; pm = 38.7511°; wcp = 24.4514 1/s; wcg = 1.2476 1/s

由以上数据可知：

剪切频率	ω_c = 1.2476 1/s
相角稳定裕度	γ =38.7511°
－π 穿越频率	ω_g =24.4514 1/s
幅值稳定裕度	$L_h=20\log(297.044483)$=49.4565 dB

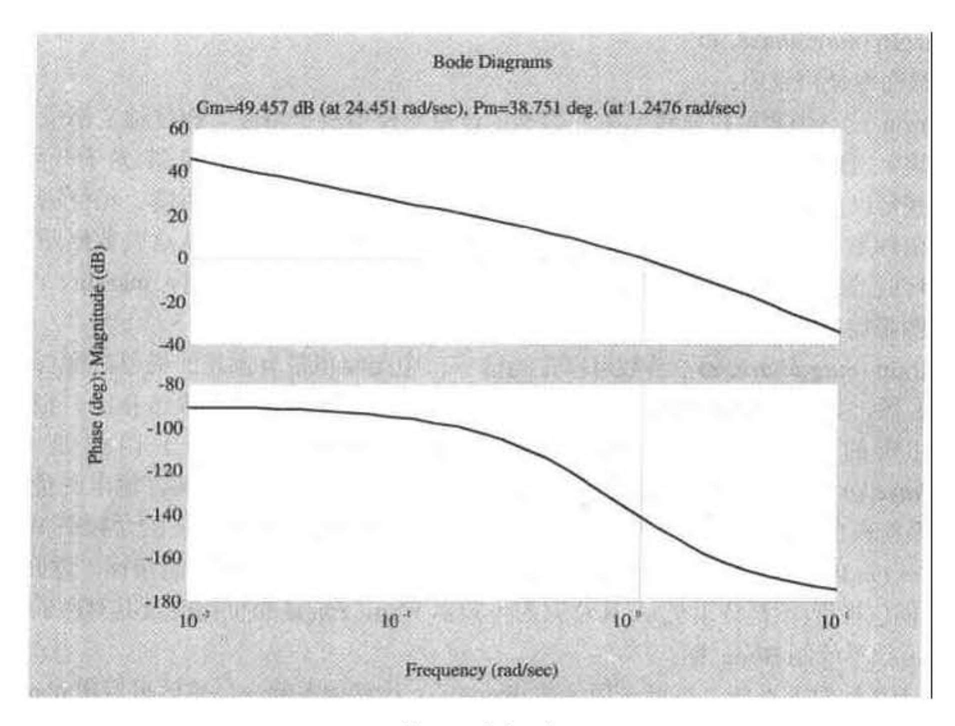


图 4-41　单位反馈系统的 Bode 图

4.5.3.2　MATLAB 命令窗口里函数频域仿真举例

【例 4-35】已知一带延迟因子系统的开环传递函数模型为：

$$\Phi(s)=10\frac{s+5}{(s+3)^2}e^{-0.8s}$$

试求其有理传递函数的频率响应，同时在一张图上绘制出以四阶 Pade 近似带延迟因子系统的频率响应，并求近似系统的频域性能指标。

【解】

根据题目要求，用函数命令编写 MATLAB 程序 L4507.m 如下：

```
%MATLAB PROGRAM L4507.m
%
n1=[10 50];d1=conv([1 3],[1 3]);s1=tf(n1,d1);
w=logspace(-1,2);[m1,p1]=bode(n1,d1,w);
t=0.8;[n2,d2]=pade(t,4);
numt=conv(n1,n2);dent=conv(d1,d2);s2=tf(numt,dent);
[m2,p2]=bode(numt,dent,w);
subplot(211);
semilogx(w,20*log10(m1),w,20*log10(m2),'r--');
subplot(212);
semilogx(w,p1,w,p2,'r--');
```

```
subplot(211);grid on
title('Bode Plot');
xlabel('Frequency (rad/sec)');
ylabel('Gain dB');
subplot(212);grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase deg');
[gm1,pm1,wcp1,wcg1]=margin(s1)
[gm2,pm2,wcp2,wcg2]=margin(s2)
```

运行该程序可得有理传递函数系统与延迟系统的 Bode 图（见图 4-42）及两系统的性能指标：

*gm*1 = Inf ；*pm*1 = 96.6256 ° ；*wcp*1 = NaN；*wcg*1 = 10.2485 1/s

*gm*2 = 1 ；*pm*2 = 0 ° ；*wcp*2 = NaN；*wcg*2 = NaN

由图 4-42 可知：对数幅频特性曲线两者重合，这是因为延迟因子对振幅不产生影响。但是在图中下边的纵坐标即对数相频特性曲线上两者有很大差别：上边一条曲线为有理传递函数对应系统的，下边的一条曲线为带延迟因子系统的 。可见延迟环节使系统的相角大大滞后，对系统产生特别大的不利影响。

由程序运行计算的数据结果也印证了从 Bode 图上所得的结论。

有理传递函数对应系统的性能指标：

剪切频率 ω_c = 10.2458 1/s　　相角稳定裕度 γ =96.6256°

-π 穿越频率 ω_g =NaN(不定值)　　幅值稳定裕度 L_h =Inf(∞)

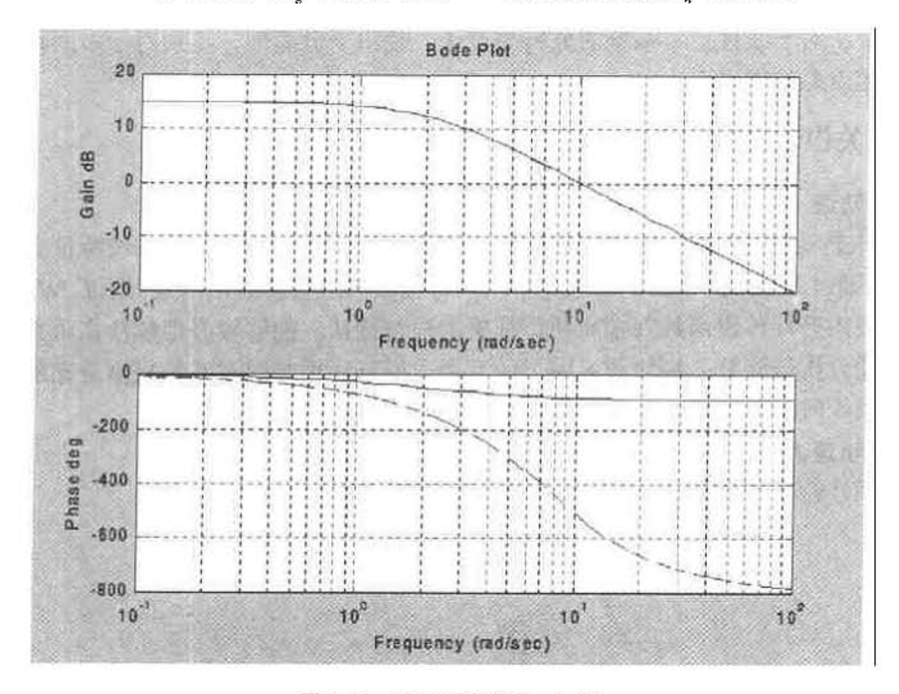


图 4-42　延迟系统的 Bode 图

带延迟因子系统的性能指标：

剪切频率 ω_c = NaN(不定值)　　相角稳定裕度 γ =0°

-π 穿越频率 ω_g =NaN(不定值)　　幅值稳定裕度 L_h =0 dB

4.6 根轨迹分析的 MATLAB 实现

若能求解系统闭环特征方程的根，就可以对系统的稳定性、快速性与准确性进行计算与分析。但是，在 20 世纪 40 年代，求解系统闭环特征方程的根，特别是求高阶系统特征方程的根还不是一件容易的事。还有，当控制系统某些参数发生变化时，需要做大量繁琐的重复计算。当然，现代科学技术特别是计算机技术的高速发展，应用计算机及其各种软件求解高次方程的根并不算什么很困难的事情。

1948 年 Evans 提出的根轨迹法是利用反馈系统中开、闭环传递函数之间的关系，由开环传递函数直接寻求闭环根轨迹的总体规律，而不去求解高阶系统的特征根。按 Evans 根轨迹法的规则，理论上可绘制出系统的根轨迹图。而绘制根轨迹图毕竟是非常繁琐的事情，绘图过程中甚至还要求解高次方程。

在 MATLAB 中，系统专门提供了函数：rlocus（）用来求系统根轨迹；rlocfind（）用来计算给定根的根轨迹增益；pzmap（）用来绘制系统的零极点图等等，这些函数都能够方便、简单而快捷地绘制根轨迹或者进行有关根轨迹的计算。

根轨迹法是分析与设计线性定常系统特别有效的图解方法，它根据根轨迹法则，迅速地作出近似的根轨迹图，直观地反映系统参数变化对根分布位置的影响。

根轨迹法对于线性定常系统的分析与设计，使用十分简便，特别对于多回路系统的研究，其优点尤其突出。

4.6.1 有关根轨迹的几个基本概念

1. 根轨迹

所谓根轨迹，是指当开环系统某一参数从零变到无穷大时，闭环系统特征方程的根在 S(复)平面上移动的轨迹。由于求高阶系统特征根的解析解异常困难，所以，W.R.Evans 提出了直接由开环传递函数判别闭环特征根的根轨迹法，能够较好地解决高阶系统控制过程的性能分析与计算。根轨迹可用于研究当改变开环增益时对系统极点分布的影响，从而提供系统时域与频域响应的分析。

2. 根轨迹方程

系统闭环特征方程的根满足

$$1+G(s)H(s)=0 \tag{4-60}$$

即

$$K^*\frac{\prod_{j=1}^{m}(s-z_j)}{\prod_{i=1}^{n}(s-p_i)}=-1 \tag{4-61}$$

此式叫做系统的根轨迹方程。

根轨迹方程式中 K^* 是系统根轨迹增益，与开环增益 K 成正比。z_j 是开环传递函数的零点。p_i 是开环传递函数的极点。

3．根轨迹法则

根轨迹法有多条规则：n 阶系统有 n 条根轨迹；根轨迹对称于实轴；根轨迹起始于开环极点，终止于开环零点与无穷远（其中 m 条终止于开环零点，n-m 条终止于无穷远）；实轴上根轨迹所在区段的右侧，开环零、极点数目之和为奇数；根轨迹渐近线方位可以计算确定；根轨迹的起始角与终止角可以计算确定；根轨迹的分离角与汇合角可以计算确定；根轨迹与虚轴的交点可以计算确定；系统 n 个开环极点之和等于系统 n 个闭环极点之和等。

4．主导极点

闭环极点中距离虚轴最近，附近又无零点的实数极点或共轭复数极点，对系统动态性能的影响最大，起主要的决定性作用，则称它们为主导极点。

5．偶极子

当某个闭环极点与某个闭环零点相距很近，其间的距离比它们到虚轴的距离小一个数量级时，则称这对零极点为偶极子。

6．系统零极点图

在复平面图上同时绘制出系统的零点与极点，极点用“×”表示，零点用“○”表示。

4.6.2 自动控制系统根轨迹分析 MATLAB 实现的主要问题

总体上说，根轨迹法所要解决的根本问题，还是系统控制过程的性能分析与计算。但具体地讲，根轨迹法是专门研究用 Evans 提出的根轨迹法规则，绘制出系统的根轨迹图，利用根轨迹图这个工具，再来对系统进行分析研究。

人工绘制根轨迹图非常麻烦与繁琐，费时费力，劳动强度大，又不易画准确。利用 MATLAB 提供的绘制根轨迹的函数：绘制系统零极点图的函数 pzmap ( )、求系统根轨迹的函数 rlocus（）、计算系统根轨迹增益函数 rlocfind（）即可简单方便、快速高效地绘制出根轨迹图，甚至有些函数例如 rltool()还带有更好、更强的其他功能。这就是控制系统根轨迹 MATLAB 实现的主要问题。

4.6.3 MATLAB 函数指令方式下根轨迹分析的 MATLAB 实现

应用 MATLAB 提供的绘制根轨迹的函数与其他函数命令、语句，编制成 MATLAB 程序，MATLAB 的这种指令方式下进行仿真是最常用的实现方法。

4.6.3.1 根轨迹 MATLAB 仿真的函数指令格式

1．绘制系统零极点图的函数 pzmap ( )

函数命令调用格式：

[*p*,*z*] = pzmap (*a*, *b*, *c*, *d*)

[*p*,*z*] = pzmap (*sys*)

[*p*,*z*] = pzmap (*p*,*z*)

函数命令使用说明：

pzmap ( ) 函数命令可以绘制线性时不变系统（LTI）的零极点图。当不带输出变量引用时，pzmap ( ) 函数可在当前图形窗口中绘出系统的零极点图。当带有输出变量引用函数时，可返回系统零极点位置的数据，而不直接绘制零极点图，如果需要可以再用 pzmap (*p*,*z*) 绘制零极点图。

pzmap (***a***, ***b***, ***c***, ***d***) 函数可以在复平面内绘制用状态空间模型描述系统的零极点图，对于 MIMO 系统，可绘制所有输入到输出间的传递零点。在图中，极点用"×"表示，零点用"○"表示。

pzmap (*sys*) 函数可以在复平面里绘制以传递函数模型 sys 表示开环系统的零极点。传递函数模型 sys 即

$$G(s)=\frac{\boldsymbol{num}(s)}{\boldsymbol{den}(s)}$$

pzmap (*p*,*z*) 函数可在复平面里绘制零极点图，其中行矢量 ***p*** 为极点位置，列矢量 *z* 为零点位置。这个函数命令用于直接绘制给定的零极点图。

【例 4-36】设一高阶系统开环的传递函数为：

$$G(s)=\frac{0.0001s^3+0.0218s^2+1.0436s+9.3599}{0.0006s^3+0.0268s^2+0.06365s+6.2711}$$

试绘制该系统的零极点图。

【解】

根据题目要求，调用 pzmap ( ) 函数命令编写如下 MATLAB 程序 L4601.m：

```
%MATLAB PROGRAM L6401.m
%
n1=[0.0001 0.0218 1.0436 9.3599];
d1=[0.0006 0.0268 0.6365 6.2711];
sys=tf(n1,d1);
pzmap(sys)
[p,z]=pzmap(sys)
```

该程序执行后可得高阶连续系统的零极点图，如图 4-43 所示。

执行程序后还计算出 3 个极点与 3 个零点：

```
p =
    -13.3371+20.0754i
    -13.3371-20.0754i
    -17.9925
z =
  -154.2949
```

```
-52.0506
-11.6545
```

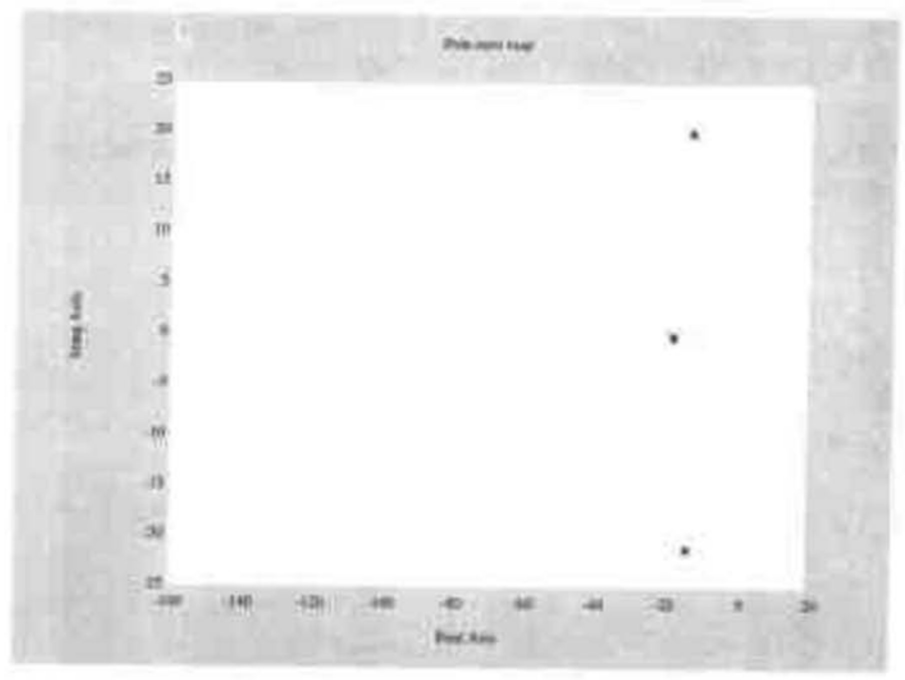


图 4-43　高阶系统的零极点图

2. 求系统根轨迹的函数 rlocus（）

函数命令调用格式：

[*r*,*k*] = rlocus (*a*, *b*, *c*, *d*)

[*r*,*k*] = rlocus (sys)

[*r*,*k*] = rlocus (*a*, *b*, *c*, *d*, *k*)

[*r*,*k*] = rlocus (*num*,*den*, k)

函数命令使用说明：

rlocus（）函数命令用来绘制 SISO 系统的 Evans 根轨迹图。给定前向通道传递函数 $G(s)$, 反馈补偿为 $k*F(s)$ 的受控对象，其闭环传递函数为

$$\Phi(s)=\frac{G(s)}{1+kG(s)F(s)}=\frac{G(s)}{Q(s)}$$

当不带输出变量引用（语句等号左端部分）时，rlocus（） 函数可在当前图形窗口中绘出系统的根轨迹图。rlocus （）函数既可适用于连续时间系统，也适用于离散时间系统。

rlocus (***a***, ***b***, ***c***, ***d***) 函数可以在复平面内绘制出用状态空间模型描述的连续或离散 MIMO 系统的根轨迹图，对于系统，增益 *k* 是自动选取的。

rlocus (***num***,***den***) 函数可以在复平面里绘出 $Q(s)$=1+k**num*(s)*den*(s)=0 闭环特征方程的根轨迹，增益 *k* 也是自动选取的。

$[r,k]$ = rlocus (a, b, c, d, k) 或 $[r,k]$ = rlocus (***num***,***den***, k) 可以用指定的 k 来绘制系统的根轨迹图。这种带有输出变量的引用函数，可返回系统复根轨迹的位置矩阵 r 及其相应的增益矢量 k，而不直接绘制出零极点图，如果需要可以再用 plot(r, 'x') 绘制出根轨迹图。

【例 4-37】续【例 4-36】，试绘制出该系统闭环的根轨迹图。

【解】

根据题目要求，用 pzmap（）函数命令编写如下 MATLAB 程序 L4602.m：

```
%MATLAB PROGRAM L4602.m
%
n1=[0.0001 0.0218 1.0436 9.3599];
d1=[0.0006 0.0268 0.6365 6.2711];
sys=tf(n1,d1);
rlocus(sys)
```

该程序执行后可得高阶连续系统的根轨迹图，如图 4-44 所示。

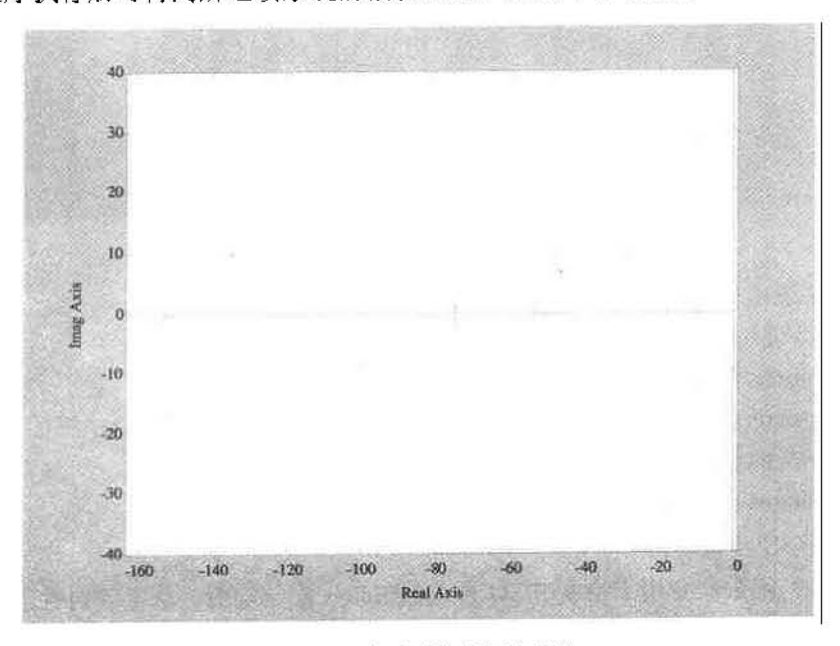


图 4-44　高阶系统的根轨迹图

3. 求给定一组根的系统根轨迹增益函数 rlocfind（）

函数命令调用格式：

[*k*,*poles*] = rlocfind (*sys*)

[*k*,*poles*] = rlocfind (*sys*, *p*)

函数命令使用说明：

rlocfind（）函数命令可计算出与根轨迹上极点（位置为 *poles*）相对应的根轨迹增益 (k)。rlocfind（）函数既适用于连续时间系统，也适用于离散时间系统。

[*k*,*poles*] = rlocfind (*sys*) 函数输入参量 *sys* 可以是由函数 tf ()、zpk ()、ss () 中任何一个建立的 LTI 对象模型。函数命令执行后，可在根轨迹图形窗口中显示十字形光标，当用户

选择根轨迹上一点时，其相应的增益由 k 记录，与增益相关的所有极点记录在 *poles* 中。

[*k*,*poles*] = rlocfind (*sys*, ***p***) 函数可对指定根计算对应的增益与根矢量 ***p*** 。

【例 4-38】已知一个单位负反馈系统开环传递函数为：

$$G(s)=\frac{k}{s(0.5s+1)(4s+1)}$$

试绘制系统闭环的根轨迹图；并在根轨迹图上任选一点，试计算该点的增益 k 及其所有极点的位置。

【解】

根据题目要求，调用 rlocfind () 函数命令编写 MATLAB 程序 L4603.m 如下：

```
%MATLAB PROGRAM L4603.m
%
n1=1;d1=conv([1 0],conv([0.5 1],[4 1]));s1=tf(n1,d1);
rlocus(s1)
[k,poles]=rlocfind(s1)
```

该程序执行后可得单位反馈系统的根轨迹图，如图 4-45 所示，并在根轨迹图窗口上显示十字形光标，当在复平面纵坐标与根轨迹交点附近选择一点时，用鼠标左键点击该点，其相应的增益由变量 k 记录，与增益相关的所有极点记录在变量 poles 中。这些数据是：

```
k =

        2.4587

poles =

       -2.2685
     0.0092394 +    0.73609i
     0.0092394 -    0.73609i
```

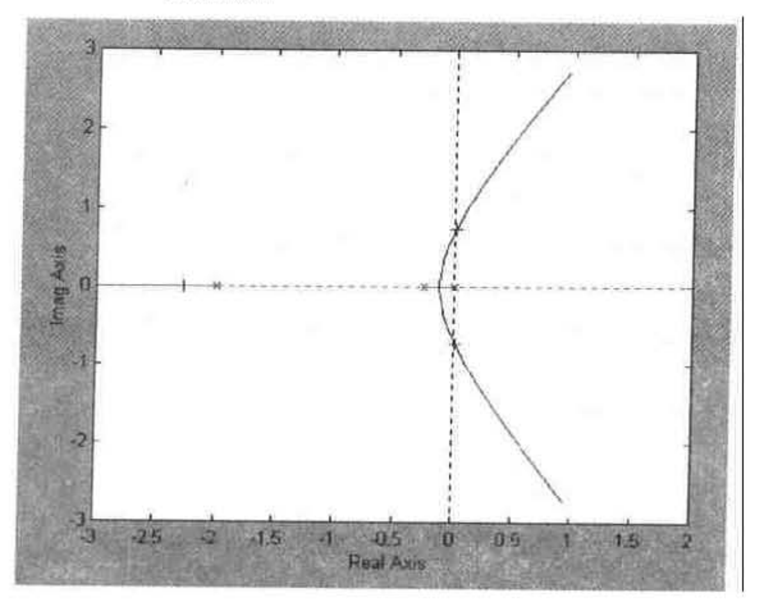


图 4-45 单位负反馈系统的根轨迹图

由程序运行计算的数据可以得知，在复平面纵坐标与根轨迹交点附近的某点，其相应的增益为 k=2.4587；与所选择交点相应的两个极点分别为：p1=(0.0092394+0.73609i)；p2=（0.0092394-0.73609i ）。

根据函数命令 rlocfind (*sys*, ***p***)，可对指定根计算对应的增益与根矢量 ***p***。如执行以下语句：

```
n1=1;d1=conv([1 0],conv([0.5 1],[4 1]));s1=tf(n1,d1);
[k,poles]=rlocfind(s1,0.0092394+0.73609i)
```

可得指定根为（0.0092394+0.73609）的增益与极点：

```
k =
         2.4586
poles =
       -2.2685
        0.0092 + 0.7361i
        0.0092 - 0.7361i
```

4．在连续系统根轨迹图与零极点图中绘制出阻尼系数与自然角频率的函数 sgrid()

函数命令调用格式：

sgrid

sgrid (*z*, *wn*)

函数命令使用说明：

sgrid ( ) 函数命令可在连续系统的根轨迹或零极点图上绘制出栅格线，栅格线由等阻尼系数与等自然振荡角频率构成。阻尼系数线步长为 0.1，范围 从 ς =0 到 $\varsigma = 1$ 。自然振荡角频率步长 为 1 rad/s，范围 从 wn =0 到 $wn = 10$。在绘制栅格线之前，当前窗口必须有连续时间系统的根轨迹或零极点图，或者是该函数必须与函数 pzmap ( ) 或者 rlocus ( )一起使用。

sgrid (*z*, *wn*) 函数可以指定阻尼系数 z 即 ς 与自然振荡角频率 wn。

【例 4-39】设一高阶系统开环的传递函数为：

$$G(s)=\frac{0.0001s^3+0.0218s^2+1.0436s+9.3599}{0.0006s^3+0.0268s^2+0.06365s+6.2711}$$

试绘制出该系统带栅格线的零极点图与带栅格线的根轨迹图。

【解】

根据题目要求，调用 sgrid（）与 pzmap（）函数命令编写 MATLAB 程序 L4604a.m 如下：

```
%MATLAB PROGRAM L4604a.m
%
n1=[0.0001 0.0218 1.0436 9.3599];
d1=[0.0006 0.0268 0.6365 6.2711];
s1=tf(n1,d1);pzmap(s1)
sgrid
```

该程序执行后可得高阶连续系统带栅格线的零极点图，如图 4-46 所示。

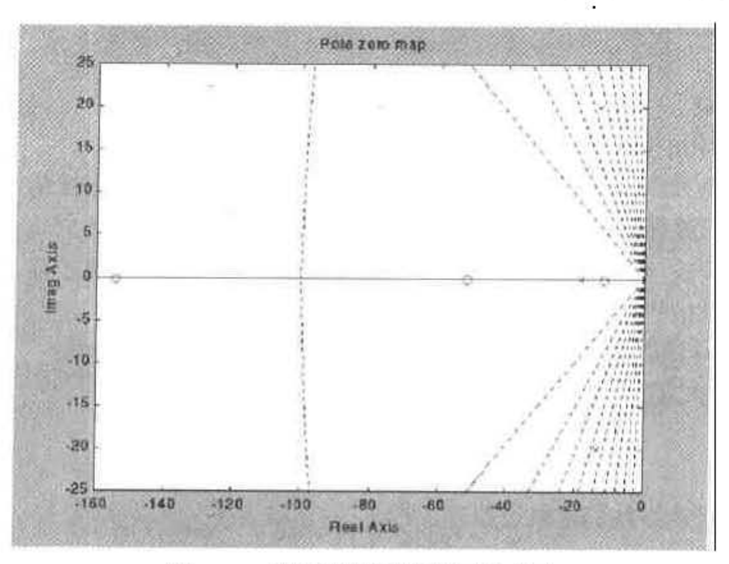


图 4-46　高阶系统带栅格的零极点图

根据题目要求，用 sgrid ( ) 与 rlocus ( ) 函数命令编写 MATLAB 程序 L4604b.m 如下：

```
%MATLAB PROGRAM L4604b.m
%
n1=[0.0001 0.0218 1.0436 9.3599];
d1=[0.0006 0.0268 0.6365 6.2711];
s1=tf(n1,d1);rlocus(s1)
sgrid
```

该程序执行后可得高阶连续系统带栅格线的根轨迹图，如图 4-47 所示。

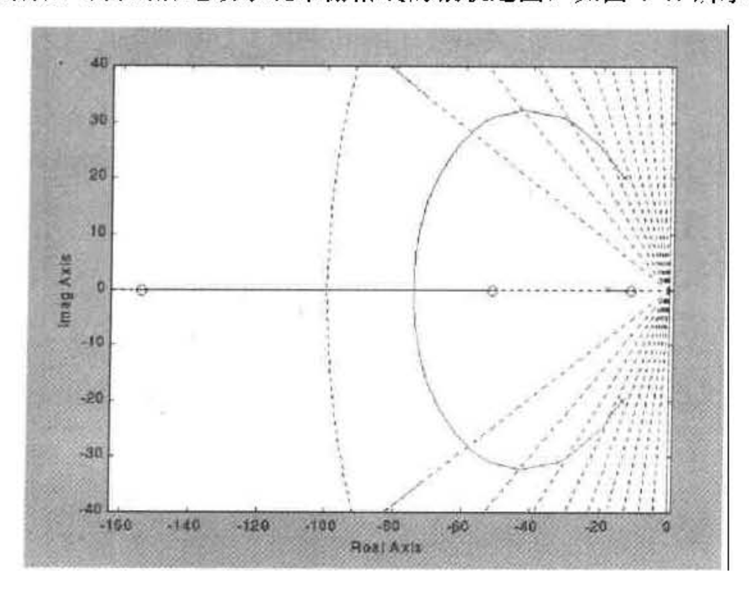


图 4-47　高阶系统带栅格的根轨迹图

5. 在离散系统根轨迹图与零极点图中绘制出阻尼系数与自然角频率的函数 zgrid ( )

函数命令调用格式：

zgrid

zgrid (*z*, *wn*)

函数命令使用说明：

zgrid ( )函数命令可在离散系统的根轨迹图或零极点图上绘制出栅格线，栅格线由等阻尼系数与等自然振荡角频率构成，阻尼系数线以步长 0.1 从$\varsigma=0$ 到 $\varsigma=1$ 绘出；自然振荡角频率以步长 $\pi/10$ 从 0 到 π 绘出。需要注意，在绘制栅格线之前，当前窗口必须有离散时间系统的根轨迹或零极点图。

zgrid (*z*, *wn*) 函数可以指定阻尼系数 *z* 即ς 与自然振荡角频率 *wn*。

【例 4-40】已知二阶离散系统的脉冲传递函数为：

$$G(z)=\frac{0.7z+0.06}{z^2-0.5z+0.43}$$

试绘制该系统带栅格线的根轨迹图。

【解】

根据题目要求，给出 MATLAB 程序 L4605.m 如下。

```
%MATLAB PROGRAM L4605.m
%
num=[0.7 0.06];den=[1 -0.5 0.43];
rlocus(num,den);
title('Root Locus')
```

该程序执行后可得离散系统带栅格的根轨迹图，如图 4-48 所示。

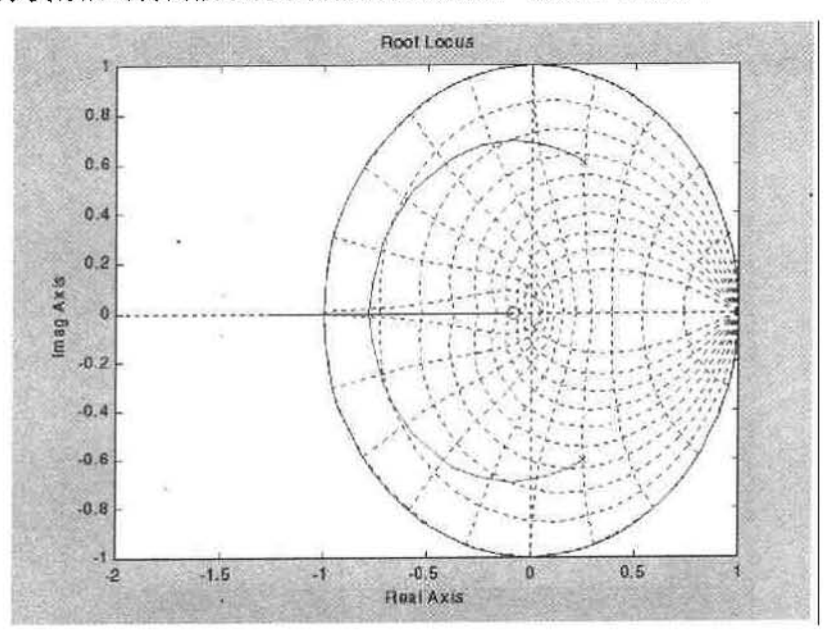


图 4-48　二阶离散系统带栅格的根轨迹图

4.6.3.2 根轨迹 MATLAB 仿真实例

【例 4-41】已知带有延迟因子的系统开环传递函数为：

$$G(s)=\frac{1}{s(s+1)(0.5s+1)}\mathrm{e}^{-s}$$

试绘制系统闭环的根轨迹图，并选择系统稳定时给定根的根轨迹增益，最后求系统 K=0.5 时的给定阶跃响应曲线。

【解】

根据题意，调用函数 rlocus ( )、rlocfind ( ) 编写如下 MATALB 程序 L4606.m：

```
% MATLAB PROGRAM L4606.m
%
n1=[1];d1=conv(conv([1 0],[1 1]),[0.5 1]);
s1=tf(n1,d1);
[np,dp]=pade(1,3);
G=s1*tf(np,dp);
rlocus(G)
[k,p]=rlocfind(G)
```

运行该程序可得系统闭环的根轨迹图，如图 4-49 所示，在图形窗口根轨迹图中显示十字形光标，当选择根轨迹负实轴上某一点时，其相应的增益由变量 k 记录，与增益相

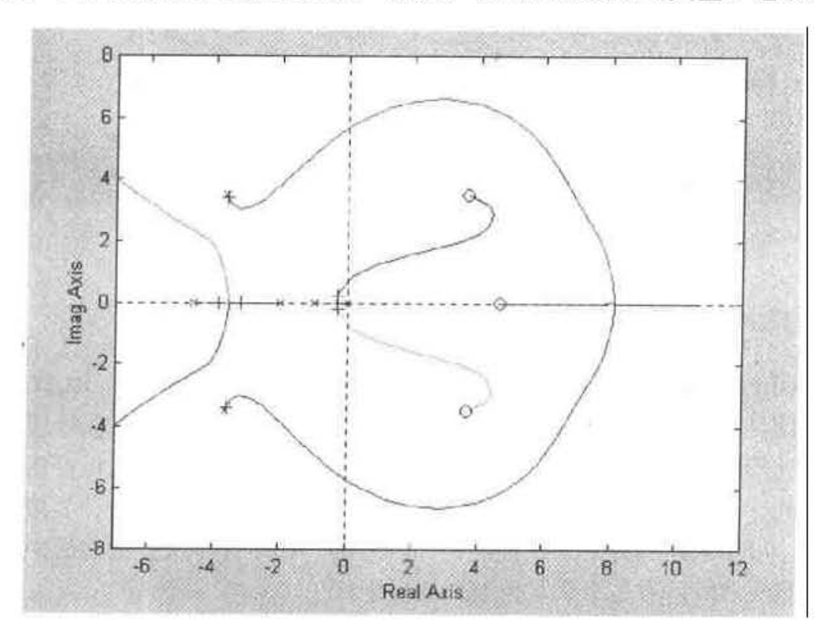


图 4-49　带有延迟因子系统的根轨迹图

关的所有极点记录在变量 poles 中。其数据如下：

```
k =
   0.1664
```

```
p =
  -3.6373 + 3.4063i
  -3.6373 - 3.4063i
  -3.8939
  -3.2383
  -0.2966 + 0.1990i
  -0.2966 - 0.1990i
```

用下面的根轨迹设计工具，可以求得当 K=0.5 时系统闭环的阶跃给定响应曲线，如图 4-50 所示。

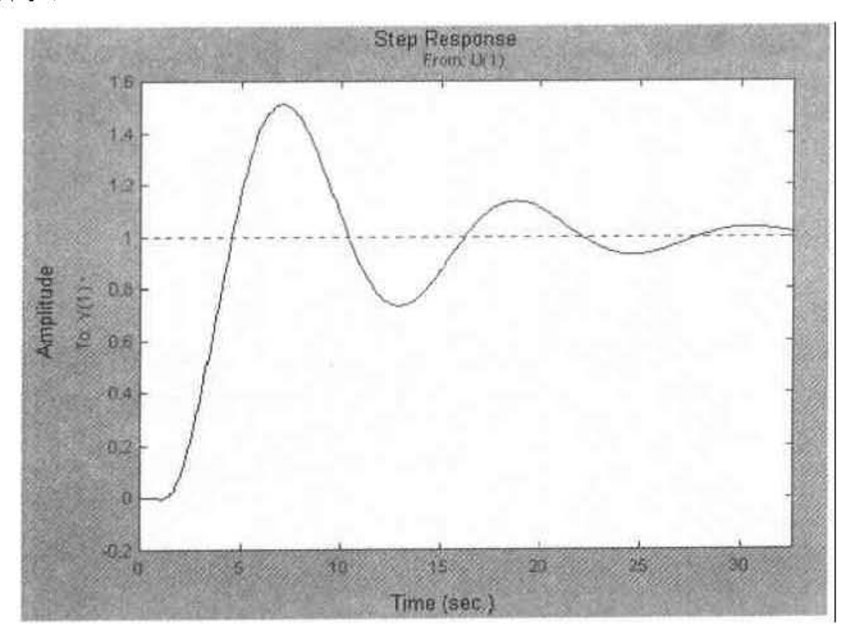


图 4-50　带有延迟因子系统的阶跃响应

4.6.4　MATLAB 基于根轨迹的设计工具

MATLAB 控制系统工具箱里有一个系统根轨迹分析与设计的工具 RLTOOL，其文件及其所在路径目录为：MATLABR11\toolbox\control\rltool.m。该文件执行后可以得到一个系统根轨迹分析的图形界面，如图 4-51 所示，在这个界面中，既可以分析系统根轨迹也能对系统进行设计。特别是能够对于被控对象在其前向通道中以设计零极点的方法来设计控制器，可用来达到提高系统控制性能的目的。在设计零极点的过程中，能够不断观察系统的响应曲线，看是否满足控制性能要求。

在图 4-51 中，最上边是标题栏“Root Locus Design”(根轨迹设计器)，第二行为菜单栏。菜单栏下面是补偿校正器 K 描述区（左）与其相应的反馈结构框图图解区（右）。K 为补偿校正器传递函数。在图解区中，K 在其前向通道里，P、H 与 F 为系统的其他部分模块，可以根据需要使用这几个模块以构造实际系统。鼠标点击各个模块可以查看、设置或者修改模块的属性。左下角的按钮用来对正负反馈进行切换选择。

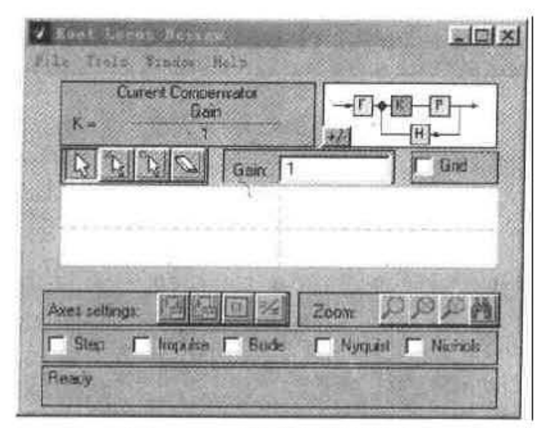


图 4-51　系统根轨迹设计器

当鼠标指向补偿校正器 K 描述区内的任何位置并点击左键时，即会弹出补偿校正器编辑对话框，如图 4-52 所示。可以在“Name”编辑框内输入补偿校正器的名称。当按下【Add Zero】按钮时，可以在“Zeros”区内的零点实部（Real）与虚部(Imaginary)编辑框内输入欲设计的零点值。当按下【Add Pole】按钮时，也可在“Poles”区内输入极点实部与虚部的值。如果输入的值不合要求，可在“Delete”框内勾选将其数据删除。补偿校正编辑器是一个非常方便有用的工具。

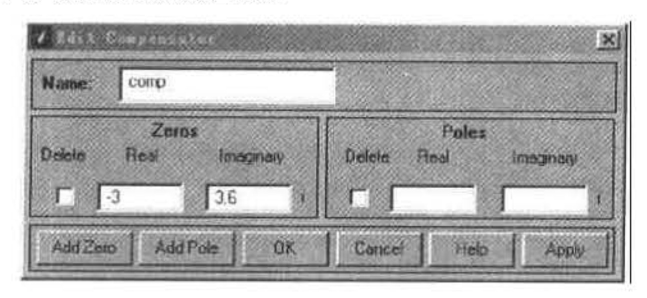


图 4-52　补偿校正器编辑对话框

在补偿器描述区与结构图解区下面是根轨迹工具栏，包括零极点添加与擦除按钮，并可以对闭环系统的零极点进行拖拽（Drag）操作。使用零极点添加与擦除按钮，可以将要构成补偿校正器的零极点放置在根轨迹图的合适地方，并不断观察系统的响应曲线，看是否满足控制性能要求。根轨迹工具栏也是一个非常直观、方便与有用的工具。

补偿器增益“Gain”编辑框内，可以改变增益值，从而改变闭环极点的位置。栅格线复选框“Grid”可以勾选根轨迹图上是否要栅格线。

设计器中间的大面积空白区是为系统绘制根轨迹之所在，当添加零极点后，也会在根轨迹图上立即显示出来，以便对系统根轨迹进行分析。绘制根轨迹区以下是坐标设置区，其按钮允许存取根轨迹绘制中的坐标轴设置。缩放区（Zoom）按钮用来对绘制的根轨迹图进行缩放显示操作。

最下边的是对系统进行各种时域与频域仿真的复选框：“Step”、“Impulse”、“Bode”、“Nyquist”、“Nichols”。当选中 5 个复选框中的任意一个或者几个时，系统立即给出相应的响应曲线。不论是一条还是多条响应曲线都在一个 LTI 视窗里。

根轨迹设计器菜单栏有 4 项菜单：【File】、【Tools】、【Window】、【Help】，每一菜单命令都有其下拉菜单，利用这些菜单命令可以对根轨迹设计器进行各种各样参数的设置，以获得设计的最佳效果。这些菜单命令大多有对应的对话框，内容比较简单，按其要求添入相关信息即可。

为使用 RLTOOL 根轨迹设计工具，MATLAB 还提供了对应功能的 rltool ( ) 函数命令。

rltool ( ) 函数的调用格式为：

rltool

rltool (sys)

rltool 函数执行后直接打开系统根轨迹设计器(图 4-51)。

rltool (sys) 函数执行后则打开系统模型 sys 带根轨迹图的设计器(图 4-53)。

【例 4-42】已知系统开环的传递函数为：

$$G(s)=\frac{(s+1)}{s(s-1)(s^2+4s-16)}$$

试用根轨迹设计器查看系统在 K=31 时的闭环单位阶跃给定响应曲线、Bode 图、单位冲激给定响应曲线；要求系统单位阶跃给定响应曲线一次微弱超调即衰减，试设计零极点校正器。

【解】

（1）显示三条曲线。根据题意，调用函数 rlocus ( ) 编写 MATALB 程序 L4607.m 如下：

```
% MATLAB PROGRAM L4607.m
%
n1=[1 1];d1=conv(conv([1 0],[1 -1]),[1 4 16]);
s1=tf(n1,d1);
rltool(s1)
```

该程序执行后，则打开本系统模型 s1 带根轨迹图的设计器，如图 4-53 所示。

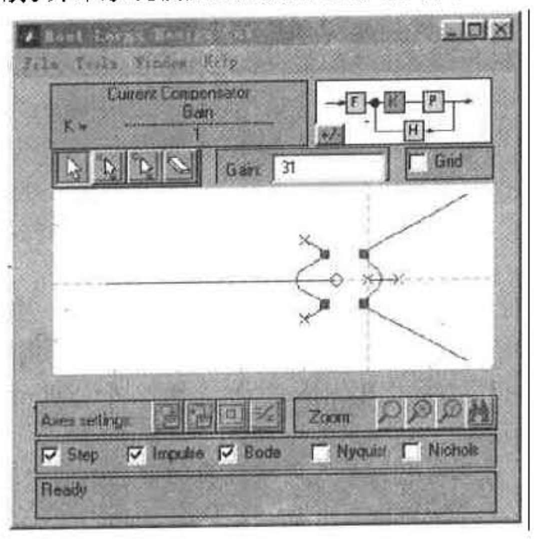


图 4-53　系统 s1 带根轨迹图的设计器

用根轨迹设计工具，可以求得当 K=31 时，系统闭环的单位阶跃给定响应曲线、Bode 图与单位冲激响应曲线三者同在一个图形窗口的特性曲线如图 4-54 所示。

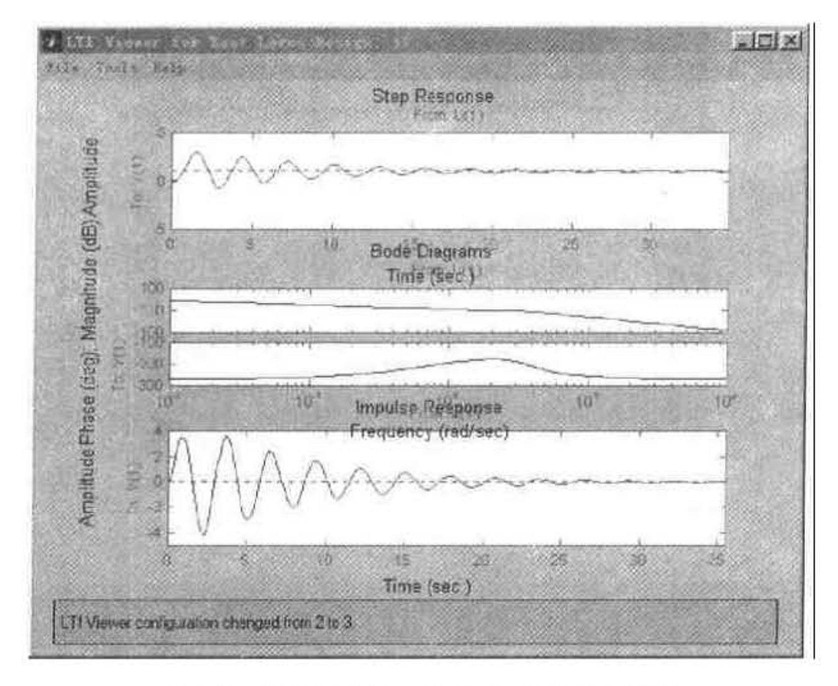


图 4-54　系统阶跃响应、Bode 图、冲激响应曲线

（2）设计零极点校正器。当用鼠标点击补偿校正器 K 描述区时，弹出补偿校正器编辑对话框，如图 4-52 所示。按下【Add Zero】按钮，在“Zeros”区内输入零点实部（-3）与虚部(3.6)，即得到一对共轭复数零点（$-3\pm3.6i$）。不需设计极点。按下【OK】或【Apply】按钮即完成了零极点校正器的设计。

经补偿校正后，带根轨迹图的设计器，如图 4-55 所示。系统闭环的单位阶跃给定响应曲线如图 4-56 所示。这样的曲线是非常理想的，调节时间短，超调小(10%左右)，并且超调后一次衰减至终了值不振荡。

【例 4-43】已知系统开环的传递函数为：

$$G(s)=\frac{1}{s^2(s+10)}$$

试用根轨迹设计器对系统进行补偿设计，使系统单位阶跃给定响应一次超调即衰减；并在根轨迹设计器中观察系统阶跃给定响应曲线与 Bode 图。

【解】

根据题意，调用函数 rlocus ( ) 编写如下的 MATALB 程序 L4608.m：

```
% MATLAB PROGRAM L4608.m
%
n1=[1];d1=conv(conv([1 0],[1 0]),[1 10]);
sys=tf(n1,d1);
rltool(sys)
```

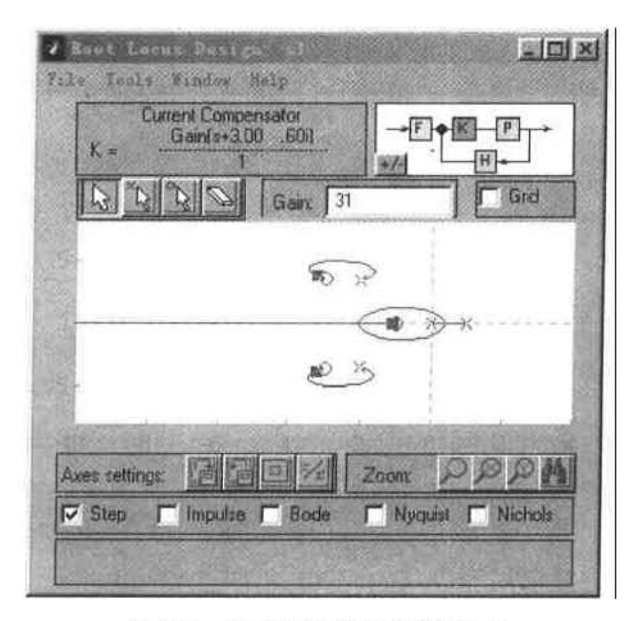


图 4-55　补偿校正后的根轨迹设计器

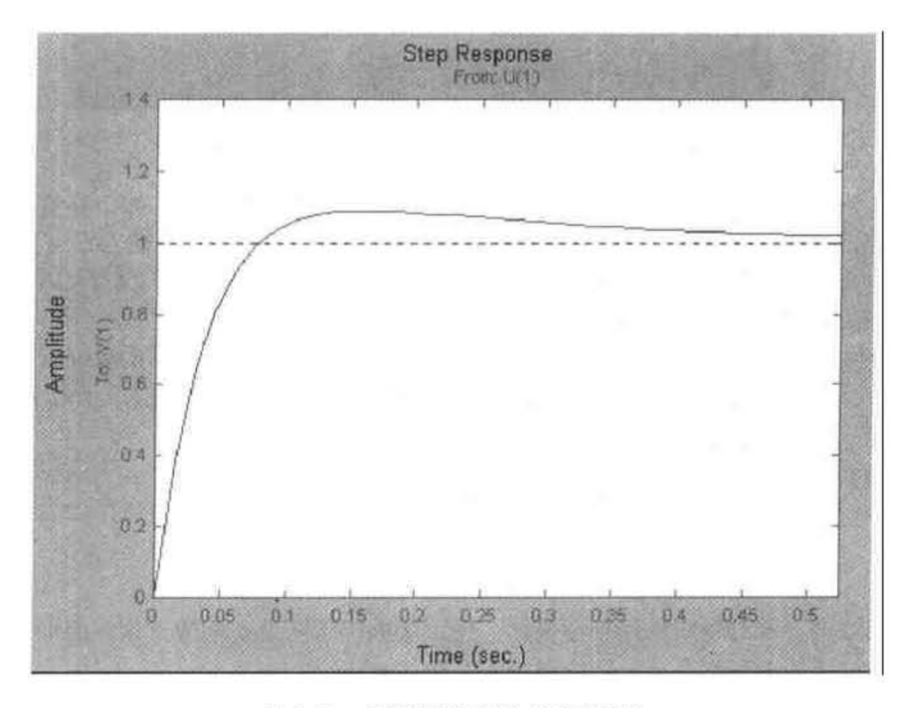


图 4-56　补偿校正后的阶跃响应曲线

函数执行后则打开系统模型 sys 带根轨迹图的设计器，如图 4-57 所示。在根轨迹设计器中，按下零点按钮用鼠标拖至根轨迹图上放在图 4-57 所示位置，即引入一个附加的负实数零点（-1.59），根轨迹图形就像图 4-57 中部图形区所绘制的形状。

当设置 K=50 时可以求得系统闭环的单位阶跃给定响应与 Bode 图两者同在一个图形窗口里的曲线，如图 4-58 所示。

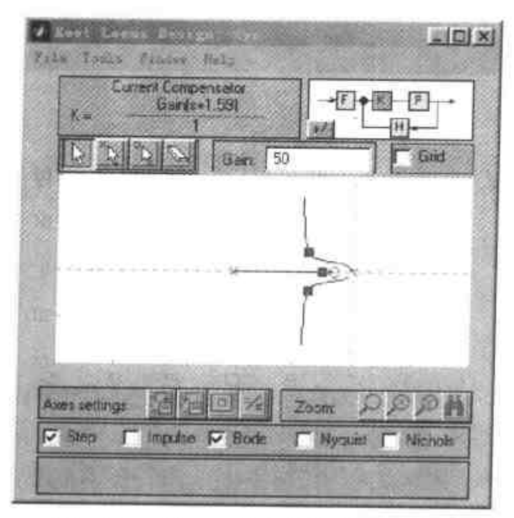


图 4-57　系统 sys 带根轨迹图的设计器

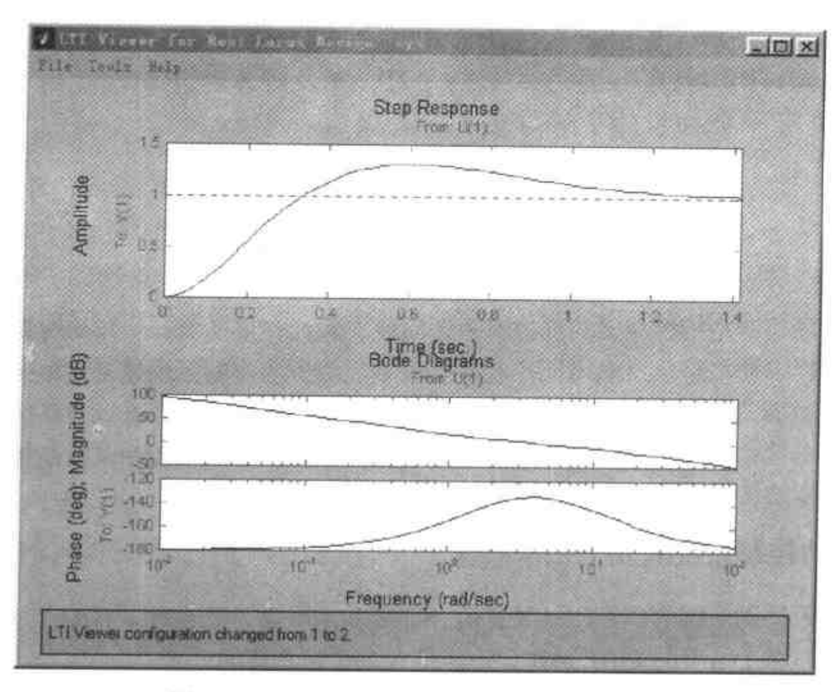


图 4-58　系统 sys 闭环阶跃响应曲线与 Bode 图

第5章　自动控制系统的 MATLAB 计算及仿真

本章在前面知识的基础上介绍控制系统的 MATLAB 计算及仿真。拖动自动控制系统与工业生产过程控制系统是控制系统系统的两大门类。本章除介绍拖动控制系统外，还要介绍过程控制系统。对于不同的系统，仍然要讨论各类系统的数学模型、各种响应的计算与仿真。本章共4节。5.1节首先介绍简单闭环控制系统的 MATLAB 计算及仿真; 5.2 节介绍多环控制系统 MATLAB 计算及仿真；5.3 节介绍位置随动系统的 MATLAB 计算及仿真; 5.4 节介绍过程控制系统的 MATLAB 计算及仿真 。通过本章的学习，可以对各种控制系统的 MATLAB 计算与仿真有一个初步的了解与基本的认识。

5.1　简单闭环控制的 MATLAB 计算及仿真

简单闭环控制是指控制器与受控对象之间既有顺向作用又有一个反馈作用的控制系统，实际工程中通常是负反馈闭环控制。工程领域的恒值控制系统通常是单闭环控制系统。例如：水塔的水位控制系统；电源恒压调节系统；电冰箱的恒温调节系统；加热炉的炉温控制系统；测速反馈的单闭环调速控制系统等等。单闭环控制系统是控制系统最重要、最基本的结构组成。

5.1.1　自动控制系统的几个基本概念

为了讨论控制系统的仿真，已经复习了许多基本概念。还有以下有关自动控制的几个基本概念，对于系统的 MATLAB 计算与仿真是非常重要的。

1．输出量与输入量

要求实现自动控制的并输送到控制对象或系统输出端上的物理量，被称为被调量或输出量。作用于受控对象或系统的输入端，并可使系统具有预定性能或预定输出的物理量，叫输入量。

2．扰动

凡是作用在系统上的破坏系统输入量与输出量之间预定规律的信号被称为扰动。

3．开环（式）控制

所谓开环控制是指控制器与受控对象之间只有顺向作用没有反馈作用的控制过程，是自动控制的基本方式之一。

4．闭环（式）控制

所谓闭环控制是指控制器与受控对象之间既有顺向作用又有反馈作用的控制过程，也是自动控制的基本方式之一。其主要特点是：

（1）闭环负反馈控制，即按偏差进行调节。

（2）抗干扰性能好，控制精度高。

（3）系统参数应适当选择，否则系统可能不能正常工作。负反馈闭环控制是自动控制中最通常采用的控制策略。

5．SISO 系统

仅有一个输入量与一个输出量的系统，即单输入单输出系统。

6．MIMO 系统

具有多个输入量与多个输出量的系统，即多输入多输出系统。

7．LTI 系统

微分方程或者差分方程的系数为常数，即不随时间变化的系统，称为线性时不变（LTI）系统。

8．镇定系统

系统输入量为常值或者是随时间缓慢变化的物理量，这样的系统多为负反馈闭环控制方式。系统的基本任务是当出现扰动时，保持系统输出量为恒定的希望值。

9．随动系统

系统输入量是随时间任意变化的物理量。这样的系统也多为负反馈闭环控制方式。系统的基本任务是使系统的输出量以要求的精度跟随输入量变化。而系统输出量通常是机械位置、速度或加速度等。

10．程序控制系统

系统输入量是按既定规律变化的物理量。系统的控制过程按预定的程序进行。系统也多为负反馈闭环控制。而系统输出量常为温度、压力、流量等物理量。

11．多闭环控制系统

所谓多闭环控制系统，是指一个闭环套一个闭环的嵌套结构组成且具有两个或两个以上闭环的控制系统。

12．电力拖动控制系统

以电动机为执行机构的控制系统。

13．过程控制系统

以化学反应或者热能转换为主要生产过程形式，以自动化仪表与装置为检测与执行机构的控制系统。

5.1.2 简单闭环控制系统的数学模型

简单闭环控制系统就是单闭环控制系统，它是所有控制系统的最基本的结构形式。其数学模型的主要形式是传递函数或零极点增益模型。已经说明过，就是系统动态结构图也是以传递函数或零极点增益模型为基础的。

若单闭环控制系统前向通道的传递函数为 $G_1(s)=\frac{\boldsymbol{num}1(s)}{\boldsymbol{den}1(s)}$，反馈通道的传递函数为

$G_2(s)=\dfrac{num2(s)}{den2(s)}$， 则闭环系统的等效传递函数为：

$$\Phi(s)=\frac{num(s)}{den(s)}=\frac{G_1(s)}{1\mp G_1(s)G_2(s)}=\frac{num1(s)den2(s)}{den1(s)den2(s)\mp num1(s)num2(s)}$$

式中，负反馈者分母取加号，正反馈者分母取减号。

在 MATLAB 中，用其函数 feedback ( ) 很容易计算并求得闭环系统的等效传递函数。

实际工程的单闭环控制系统仿真通常是通过 SIMULINK 动态结构图实现的。实现的思路是先将动态结构图转换为状态空间模型，然后再仿真。具体过程是用 SIMULINK 提供的 linmod（） 和 linmod2（）两个函数，从连续系统中提取线性模型。两个函数命令执行后，都可以得到一个用[A,B,C,D]表达的状态空间模型。然后就可以对这个状态空间模型来进行各种仿真（参见 3.7 节）。

这种方法之所以被普遍采用，是因为：其一，实际工程的控制系统，是由一个一个环节按特定控制要求连接而成，经抽象及近似处理后即得 SIMULINK 动态结构图，动态结构图就是可见的原始模型，不需要再费力费时做工作动态结构图就可以用来进行仿真；其二，除开极少数简单问题外，有了动态结构图再求传递函数，一般都较复杂、麻烦。因此，通过 SIMULINK 动态结构图模型实现仿真是较好的方法。

需要特别注意系统的 SIMULINK 动态结构图模型的.mdl 文件在计算机磁盘空间里存放的路径，并注意区别在书中的插图编号与在 MATLAB 中的 SIMULINK 结构图模型.mdl 文件名之间的对应关系。

5.1.3 简单闭环控制系统的 MATLAB 仿真实例

【例 5-1】已知晶闸管-直流电机单闭环调速系统（V-M 系统）的 SIMULINK 动态结构图如图 5-1 所示。

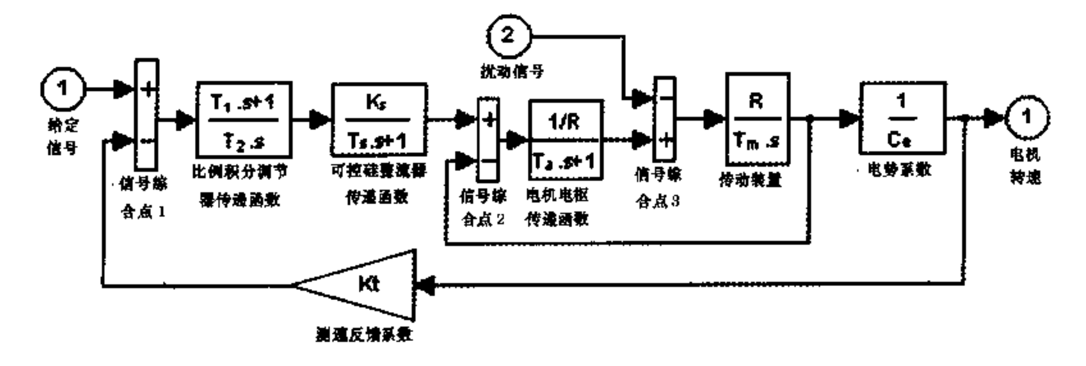


图 5-1 转速单闭环调速系统的 SIMULINK 动态结构图

图中直流电机 $P_{nom}=10\,\text{kW}$， $U_{nom}=220\,\text{V}$， $I_{nom}=55\,\text{A}$， $n_{nom}=1000\,\text{r}/\text{min}$，电枢电阻 $R_a=0.5\,\Omega$， V-M 系统主电路总电阻 $R=1\,\Omega$，额定磁通下的电机电动势转速比 $C_e=0.1925\,\text{V}\cdot\text{min/r}$，电枢回路电磁时间常数 $T_a=0.017\,\text{s}$，系统运动部分飞轮矩相应的机

电时间常数 $T_m = 0.075\,s$，系统测速反馈系数 $K_t =$ 0.01178 v/rpm，整流触发装置的放大系数 $K_s = 44$，三相桥平均失控时间 $T_s = 0.00167\,s$，比例积分调节器的两个系数 $T_1 = 0.049\,s$，$T_2 = 0.088\,s$。试绘制出该系统的单位给定阶跃响应曲线；并绘制出在图 5-1 中扰动信号作用点处的单位阶跃扰动与单位冲激扰动的响应曲线。

【解】

（1）解题的几点考虑　采用工程上惯用的做法，已将可控硅及其触发装置近似为一阶惯性环节。在图 5-1 中，内反馈为额定励磁下直流电机里的电势反馈，这是由直流电机工作原理所决定的。用测速发电机作反馈控制元件构成负反馈闭环控制，这是一个单闭环控制系统。若计及直流电机里的电势反馈闭环，则是一个双闭环系统。

请特别注意，系统动态结构图的.mdl 文件在计算机磁盘空间里存放的路径最好选择为MATLABR11\WORK，这样便可省略路径。

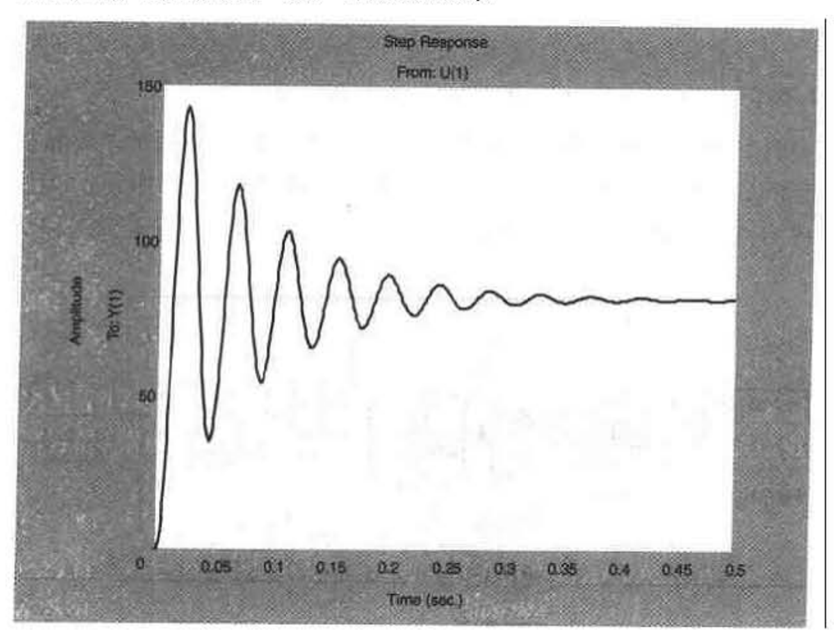
（2）根据题目要求，用 step( ) 函数命令编写求单位阶跃响应的 MATLAB 程序段如下。

```
[a,b,c,d]=linmod2('t5101');
sys=ss(a,b,c,d);
step(sys)
```

SIMULINK 动态结构图 t5101.mdl 文件只能存放在MATLAB 中，图 5-1 即是 t5101.mdl 结构图。运行以上程序段可得系统的单位阶跃响应曲线，如图 5-2 所示。

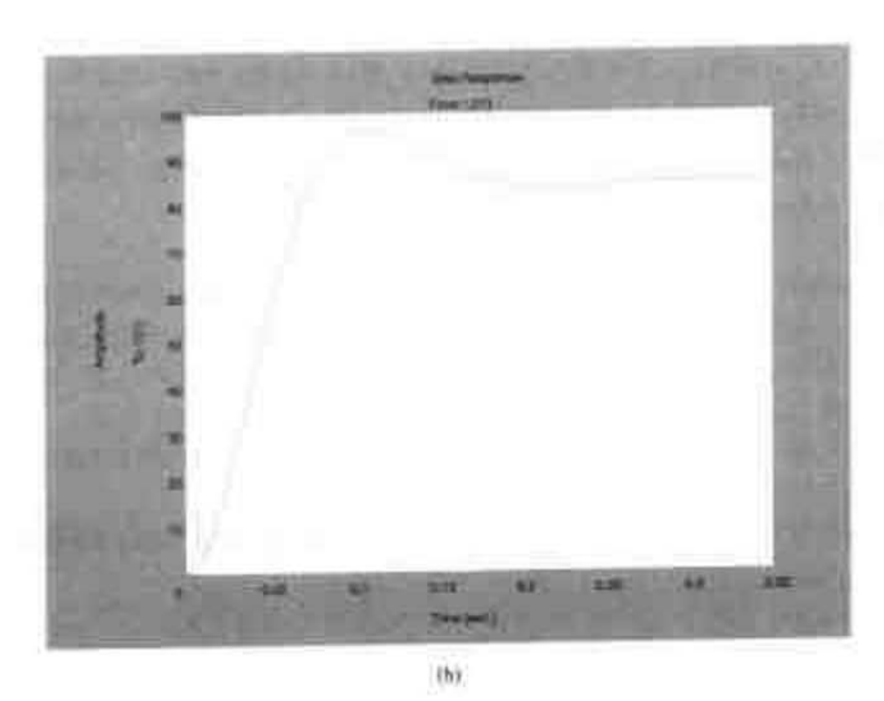
当比例积分调节器仅取为比例放大器（放大倍数 $K_p = 10$）时，单位阶跃响应曲线如图 5-2 (a) 所示，系统阶跃响应呈现为衰减振荡。

当比例积分调节器取参数 $T_1 = 0.049\,s$ 与 $T_2 = 0.088\,s$ 时，单位阶跃响应曲线如图 5-2 (b) 所示，系统阶跃响应稍微超调一点，即迅速衰减。



(a)

图 5-2



(b)

图 5-2　单闭环系统的单位阶跃响应曲线

（3）根据题目要求，调用函数命令 impuls ( ) 编写求单位阶跃扰动响应、单位冲激扰动响应的 MATLAB 程序段。

```
[a,b,c,d]=linmod2('t5103');
figure(1);
step(a,b,c,d);hold on
figure(2);
impulse(a,b,c,d)
```

程序中 t5103.mdl 结构图即图 5-3。运行该程序段可得系统的单位阶跃扰动响应曲线，如图 5-4（a）所示；单位冲激扰动响应曲线如图 5-4（b）所示。请注意设置图 5-3 中信号综合点 2 的极性，当为阶跃扰动信号时取负极性，为单位冲激扰动信号时取正极性。

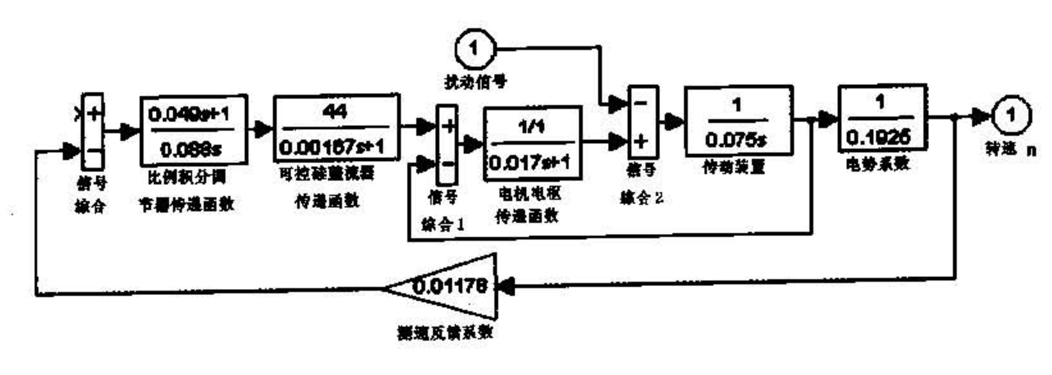
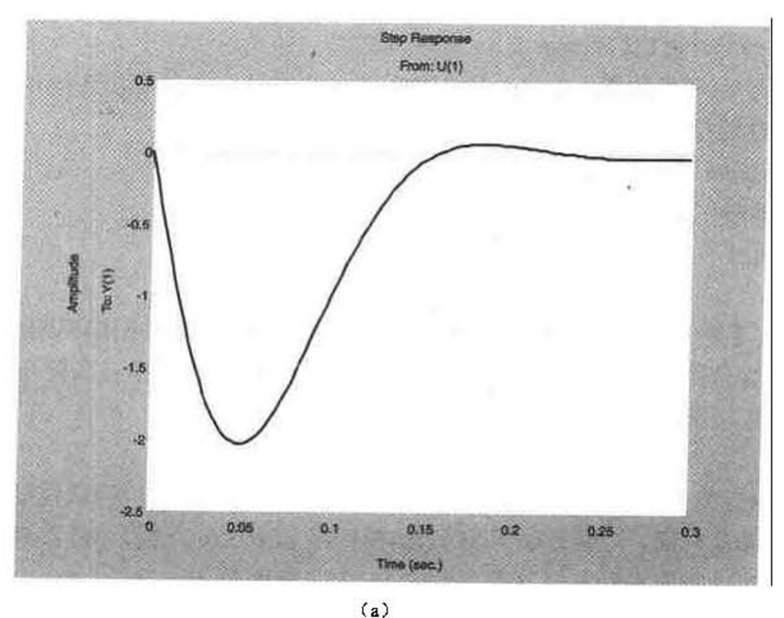
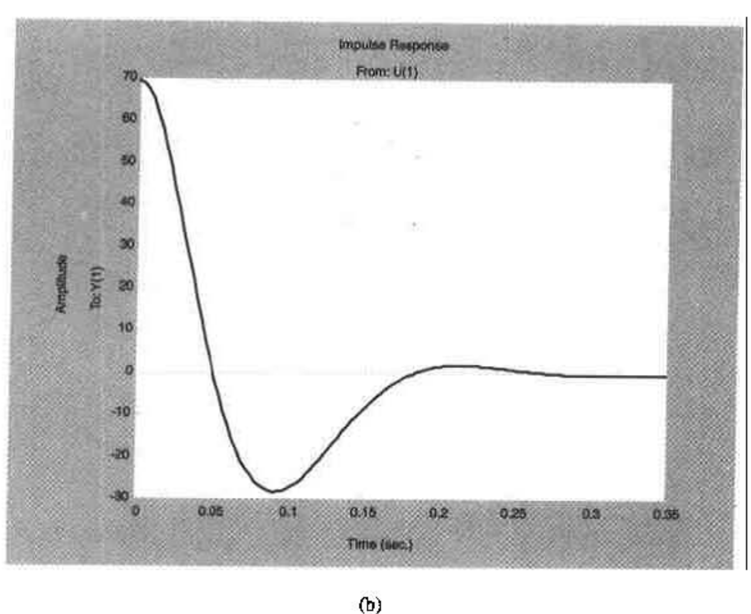


图 5-3　单闭环调速系统的扰动信号作用结构图



(a)



(b)

图 5-4 单闭环系统在给定点的单位阶跃扰动与单位冲激扰动响应曲线

【例 5-2】续【例 5-1】，试计算采用比例积分调节器的转速单闭环系统的单位阶跃响应的性能指标。

【解】

根据题目要求，调用函数命令 perf( ) 编写 MATLAB 程序段如下：

```
global y t
[a,b,c,d]=linmod2('t5101');
sys=ss(a,b,c,d);
```

```
[y,t]=step(sys);perf(1,y,t);
```

程序执行后,可得如图 5-2（b）所示的阶跃响应曲线，并计算出转速单闭环系统单位阶跃响应的性能指标如下：

超调量 *sigma* =13.7073 %

峰值时间 *tp* = 0.1094 s

调节时间 *ts* = 0.1550 s(对应于 5% 的误差带)

0.1671 s(对应于 2% 的误差带)

【例 5-3】续【例 5-1】，对于图 5-3 所示采用比例积分调节器的单闭环调速系统，试计算在图中扰动信号作用点施加单位阶跃扰动信号时的最大动态降落及其降落时间，并计算以扰动信号作用前系统输出为基准值 5%范围的恢复时间。

【解】

程序中 t5103.mdl 结构图即图 5-3。根据题目要求，求解单位阶跃扰动信号的最大动态降落及其降落时间，可编写如下 MATLAB 程序 L5103.m。该程序经整理完善可以改编为对于阶跃扰动响应求动态降落等参数的 MATLAB 函数 dist.( )。

```
% MATLAB PROGRAM L5103.m
%
[a,b,c,d]=linmod2('t5103');
step(a,b,c,d)
%Count detac tp
[y,x,t]=step(a,b,c,d);
[detac,tf]=min(y);
cs=length(t);
yss=y(cs);
detac
tp=t(tf)
%Count tv
i=cs+1;n=0;
while n==0,
        i=i-1;
       if i==1,
              n=1;
       elseif y(i)>=0.05,
              n=1;
       end;
end;
t1=t(i);
cs=length(t);j=cs+1;n=0;
while n==0,
   j=j-1;
```

```
            if j==1,
                    n=1;
             elseif y(j)<=(-0.05),
                    n=1;
                end;
end;
t2=t(j);
if t2<tp
       if t1>t2
                 tv=t1
        end
         elseif t2>tp,
if t2<t1,
     if t1~=0
            tv=t1
   else
tv=t2
end
else
  if t1~=0
           tv=t1
else
  tv=t2
end
        end
      end
```

程序执行后,可得图 5-4 (a) 所示单闭环系统的单位阶跃扰动响应曲线，并有

最大动态降：*detac*= -2.0259

最大动态降时间：*tp* = 0.0514 s

恢复时间：*tv*= 0.2003 s(对应于 5% 的误差带)

0.2165 s(对应于 2% 的误差带)

即计算出了转速单闭环系统单位阶跃扰动的最大动态降落为-2.0259 及其对应时间为 0.0514s，以扰动信号作用前系统输出为基准值 5%范围的恢复时间为 0.2003s。基准值 2%范围的恢复时间为 0.2165s。如果运行以下调用 dist.( )函数的程序段，会得到同样的结果。

```
[a,b,c,d]=linmod2('t5103');
step(a,b,c,d);
[y,x,t]=step(a,b,c,d); [detac1,tp1,tv1]=dist(1,y,t);
[detac2,tp2,tv2]=dist(2,y,t);
```

【例 5-4】续【例 5-1】，试对采用比例积分调节器的转速单闭环系统的进行频域分析。

【解】

对转速单闭环系统进行频域分析，即求其频域性能指标。可以用两种方法进行求解：一种方法是借助于系统的开环 SIMULINK 结构图，经转换成状态空间模型进行求解；另一种方法是用系统的开环传递函数求解。

（1）第一种方法

系统开环 SIMULINK 结构图如图 6-5 所示，即以下程序中的动态结构图 t5105.mdl，这样安排是为了方便进行仿真。

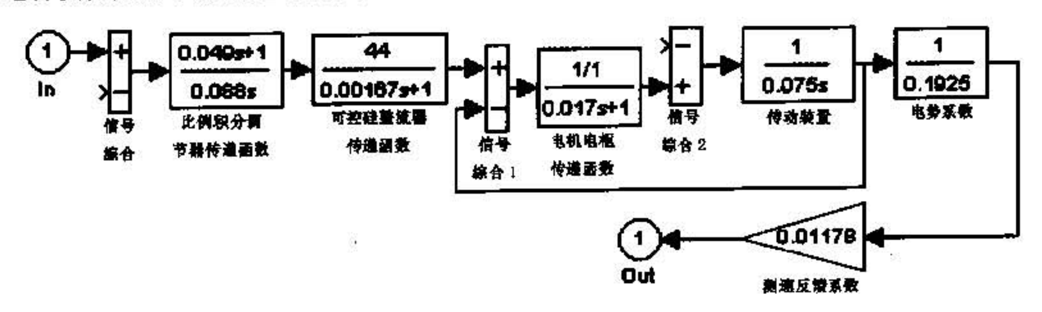


图 5-5 转速单闭环系统的开环 SIMULINK 结构图

根据题目要求，用函数命令编写如下 MATLAB 程序段。

```
[a,b,c,d]=linmod2('t5105');
sys=ss(a,b,c,d);
margin(sys)
```

程序段执行后，即可绘制出转速单闭环系统的 Bode 图，如图 5-6 所示，并计算出频域性能指标。

由图 5-6 可知系统的：模稳定裕度 G_m=26.369 dB； -π 穿越频率 ω_{cg} = 151.67 s^{-1}；

相稳定裕度 P_m=54.016 deg； 剪切频率 ω_{cp} = 25.484 s^{-1}

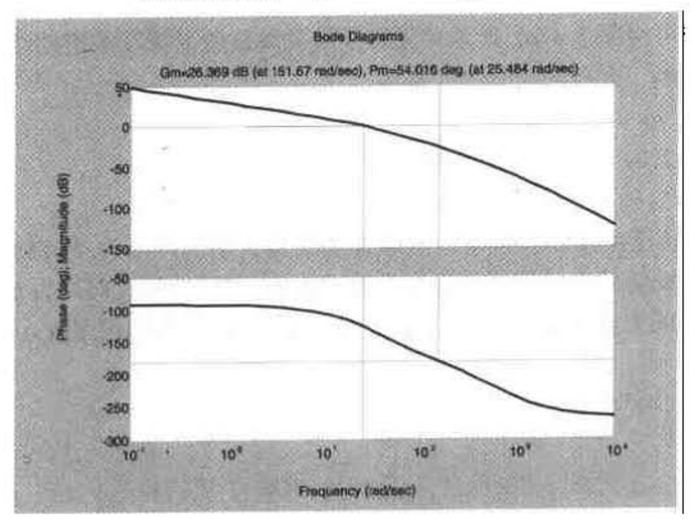


图 5-6 转速单闭环系统的 Bode 图与频域性能指标

由仿真结构图 5-5 可以看出，这样的仿真考虑了直流电机内的电势反馈。

（2）第二种方法

根据系统 SIMULINK 结构图 5-1 或图 5-3，求转速单闭环系统的开环传递函数，然后进行仿真。根据题目要求，调用函数命令 bode ( ) 与 margin ( ) 编写的 MATLAB 程序 L5104.m 如下。

```
% MATLAB PROGRAM L5104.m
%
num1=1;den1=[0.017 1];num2=1;den2=[0.075 0];
[num,den]=series(num1,den1,num2,den2);
[numc,denc]=cloop(num,den);
num3=[0.049 1];den3=[0.088 0];
num4=[0 44];den4=[0.00167 1];
num5=1;den5=0.1925;
[numa,dena]=series(numc,denc,num3,den3);
[numb,denb]=series(num4,den4,num5,den5);
[numz,denz]=series(numa,dena,numb,denb);
num6=0.01178;den6=1;
[numw,denw]=series(numz,denz,num6,den6);
[mag,phase,w]=bode(numw,denw);
margin(mag,phase,w)
```

程序执行后，即可得到与第一种方法几乎完全相同的结果。而且由仿真程序可以看出，这种基于传递函数的仿真也考虑了直流电机的电动势内反馈。

【例 5-5】已知一系统动态结构图如图 5-7（t5107.mdl）所示，试作出系统的根轨迹图。再用根轨迹设计器对系统进行零极点设计，使系统的性能得以提高。

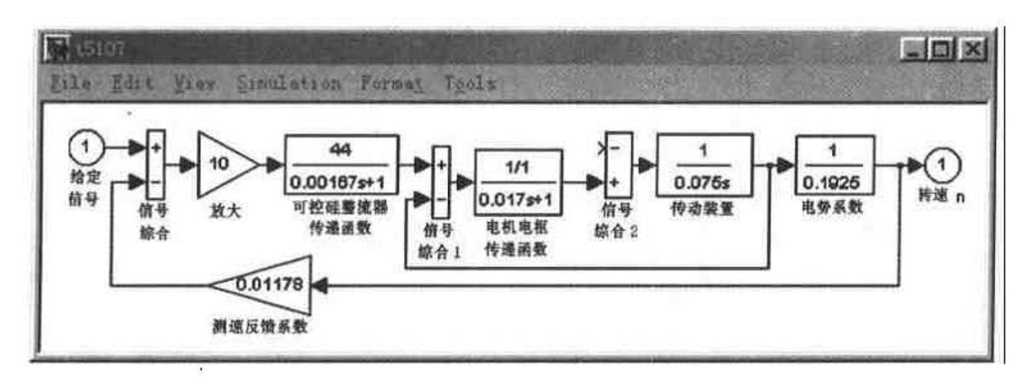


图 5-7 系统动态结构图

【解】

（1）作出系统的根轨迹图

程序中 t5107a.mdl 动态模型就是 t5107.mdl 断开闭环连接的结构图。根据题意，调用函数命令 rlocus ( ) 的 MATLAB 程序段如下：

```
[a,b,c,d]=linmod2('t5107a');
sys=ss(a,b,c,d);
rlocus(sys)
```

该程序段执行后可得系统的根轨迹图，如图 5-8 所示。

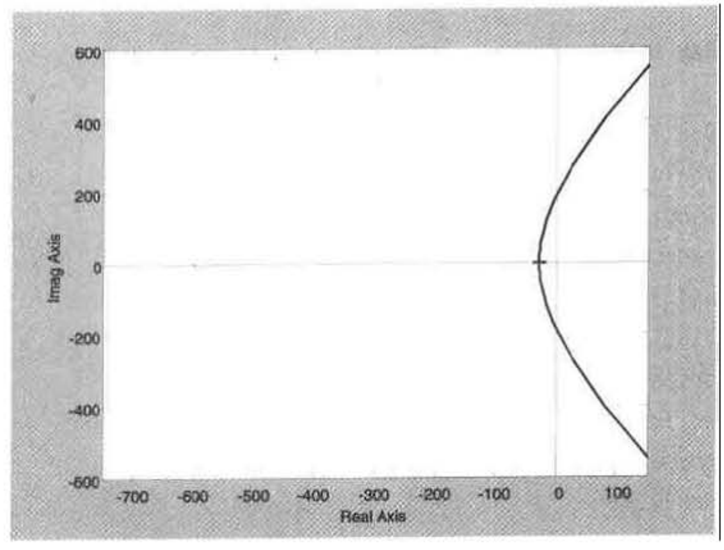


图 5-8　系统的根轨迹图

（2）用根轨迹设计器对系统进行零极点设计

原系统(闭环)的阶跃给定响应衰减振荡较严重（参看图 5-9）。根据题意，调用函数命令 rltool（ ）的 MATLAB 程序段如下。程序中 t5107a.mdl 动态模型就是 t5107.mdl 断开闭环连接的结构图，因为绘制根轨迹图需用系统的开环模型。

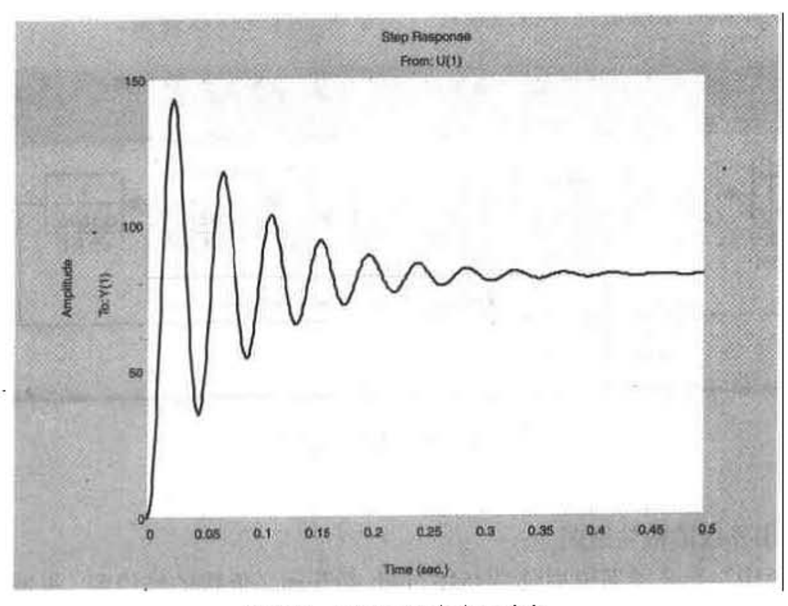


图 5-9　校正前系统阶跃响应

```
[a,b,c,d]=linmod2('t5107a');
sys=ss(a,b,c,d);
rltool(sys);
```

程序段运行后， 在根轨迹设计器（图 5-10）里，鼠标左键双击补偿校正器编辑区打开对话框，如图 5-11 所示，设计一对共轭复数零点（$-719.3548\pm214.6119i$）、另一个零点（-171）、一个极点（-10）。然后在根轨迹设计器里仿真复选框处勾选“Step”与“Nyquist”两项，即可得经补偿校正后的带根轨迹图的设计器，如图 5-10 所示。系统闭环的单位阶跃给定响应与 Nyquist 曲线如图 5-12 所示。这样的阶跃响应曲线是理想曲线，因为阶跃给定响应输出曲线几乎是呈 90° 上升的阶跃，实际系统并不能达到，多少总是会有些惯性的。

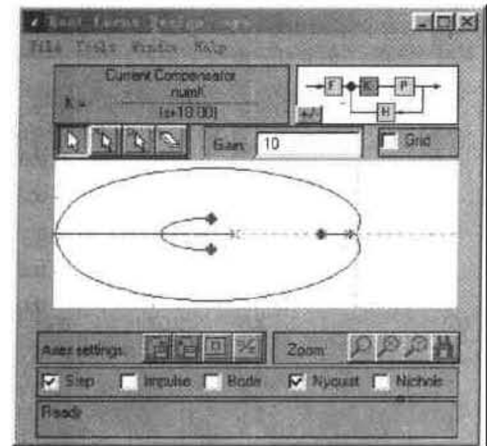


图 5-10　补偿校正后的设计器界面

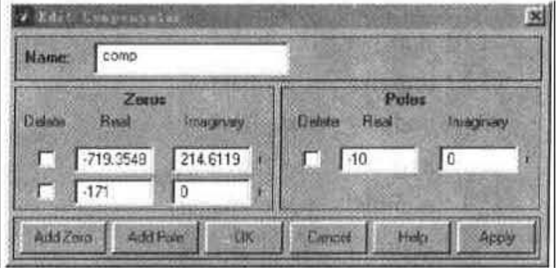


图 5-11　补偿校正器编辑对话框

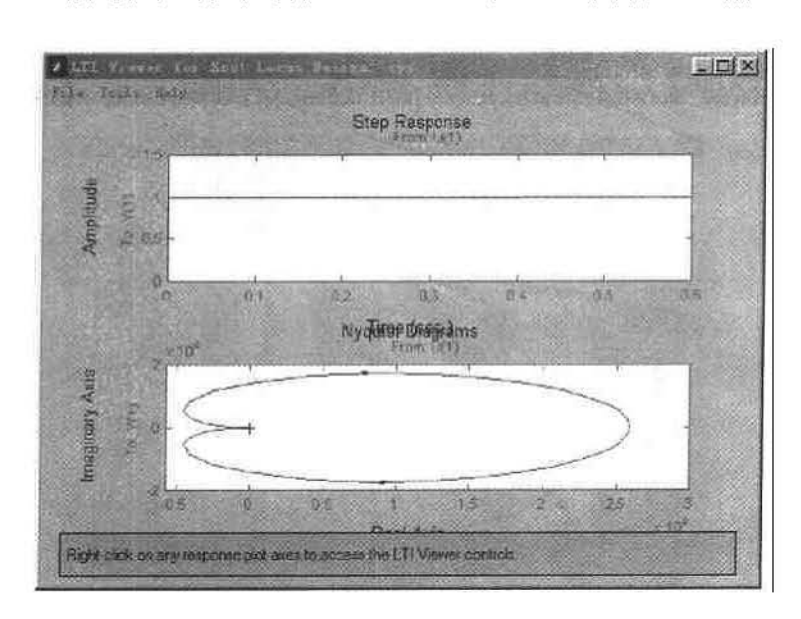


图 5-12　补偿后的阶跃给定响应与 Nyquist 曲线

5.1.4 线性时不变系统（LTI）观测器

LTI 观测器是 LTI 系统各种响应的分析工具。MATLAB5.x 的几个版本都提供了线性时不变系统（LTI）观测器，比起老版本，MATLAB5.3 有了很大改进。这个工具是通过 ltiview ( ) 函数来打开的。ltiview ( ) 函数的调用格式为：

ltiview

ltiview (*plottype*)

ltiview (*plottype*,*sys*)

ltiview (plottype,sys1,*sys*2,…*sysN*)

ltiview (plottype,sys1,plotstru1,sys2, plotstru2,…sysN, plotstruN)

函数使用说明：

ltiview 函数用来打开一个 LTI 观测分析器，其中没有任何响应曲线。

ltiview（ *plottype*）函数用来打开一个 LTI 观测分析器，其中的系统是一个随机的二输入、二输出、四状态的状态空间 LTI 对象（）。输入参量 plottype 是指代系统分析的响应类型，它可以是以下 9 种字符串之一：

（1）“step” —阶跃响应。

（2）“impulse” —脉冲响应。

（3）“bode” —Bode 图。

（4）“nyquist” —Nyquist 图。

（5）“nichols” —Nichols 图。

（6）“sigma” —奇异值响应。

（7）“pzmap” —极点零点图。

（8）“lsim” —任意信号输入的响应。

（9）“initial” —零输入响应。

ltiview('step')函数命令执行后，打开一个随机系统的 LTI 观测分析器，如图 5-13 所示。

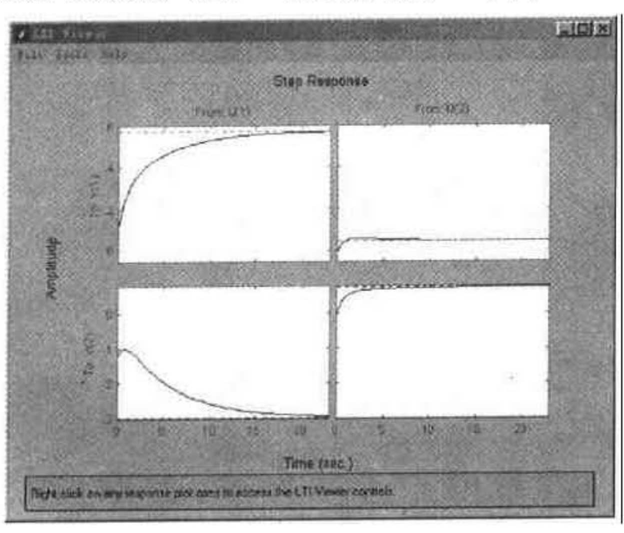


图 5-13 随机系统的 LTI 观测分析器

ltiview（ *plottype,sys*）函数用来打开一个 LTI 观测分析器，其中的系统是 sys 的 LTI 对象。输入参量 *plottype* 是响应分析的类型。

ltiview (*plottype*,sys1,*sys*2,…*sysN*) 函数用来同时分析多个系统的同一种响应。

ltiview (*plottype,sys*1,*plotstr*1,*sys*2, *plotstr*2,…*sysN*, *plotstrN*) 函数用来同时分析多个系统的同一种响应，而每个系统的响应曲线的颜色、线形、标识符号可由 plotstr 来设置。

LTI 观测器菜单栏有三个菜单项：【File】、【Tools】、【Help】，每一菜单命令都有其下拉菜单，利用这些菜单命令可以对 LTI 观测器进行各种各样参数的设置，以获得观测与分析的最佳效果。LTI 观测器对系统的各种观测与分析，主要是通过菜单操作实现的。

菜单栏下是系统响应曲线绘制区，在这里除了绘制系统各种时域响应曲线外，还绘制系统各种频域响应图与系统零极点布置图。

在主菜单的【Tools】命令下，有三个菜单项命令【Viewer Configuration】、【Response Preferences】、【Linestyle Preferences】是非常有用的。

【Viewer Configuration】菜单命令为 LTI 观测分析器可用视窗外形布置对话框，如图 5-14 所示。由图看出，LTI 观测分析器有 6 种可选择的响应曲线组合，第 1 种只有 1 条响应曲线，依次到第 6 种组合为 6 条响应曲线，均按图中安排的曲线顺序进行排列。每一条曲线都能在下拉式菜单里进行选择不同的响应类型。对话框下的 4 个按钮及其作用已在第三章的相关内容里介绍过，不再赘述。

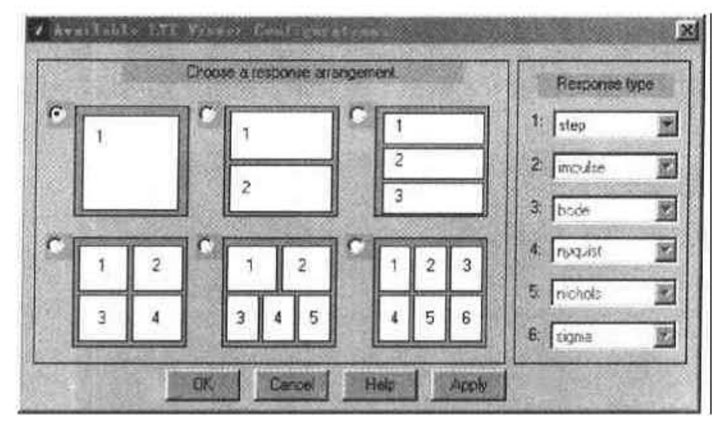


图 5-14　LTI 观测分析器可用视窗外形布置对话框

【Response Preferences】为 LTI 观测分析器响应参数选择设置对话框，如图 5-15 所示。由图可知，对话框有时域区与频域区两部分。在时域区中，其一，可以设置时间向量，既可以自动生成时间向量，也可设定时间向量，时间向量设置格式为 [起始时间：时间间隔：终止时间]；其二，可以设置 Y 轴大小范围，既可自动生成 Y 轴范围，也可由用户设置 Y 轴范围。其三，可以对时域响应曲线的调节时间与上升时间的条件进行设置。对于调节时间，系统默认 2%的误差带，可以改变设置取 5%的误差带。对于上升时间，系统默认为 10%上升到 90%的时间间隔，也可改变设置。

在频域区中，其一，可以设置频率向量，既可以自动生成频率向量，也可定义频率对数空间。对数空间设置格式为$[10^{n1},10^{n2},k]$，表示其值从10^{n1}到10^{n2}，在两者之间有 k 个点。还可以对每一个响应类型是否都进行核算进行勾选。其二，可对响应幅值的单位

进行选择：分贝 dB，绝对值，对数运算值等；也可以对响应相位角的单位进行选择：弧度 rad，度（°）；还可以对频率的单位进行选择：角频率 rad/s，频率 Hz。

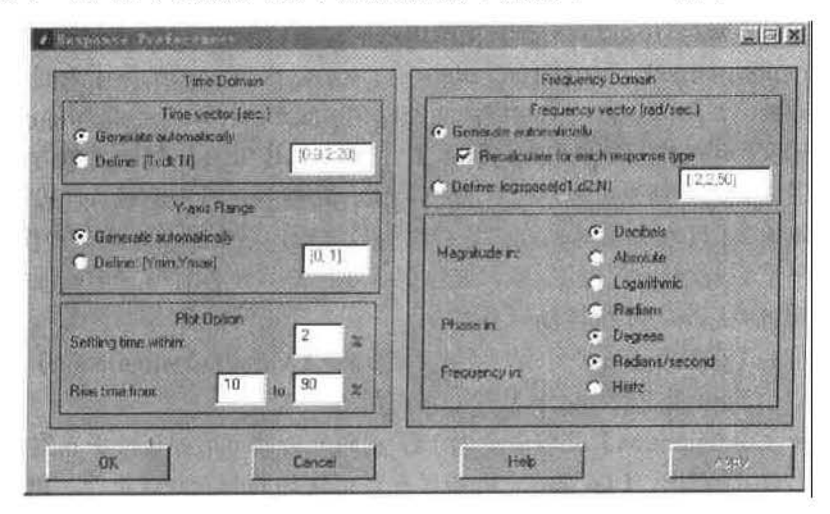


图 5-15　LTI 观测分析器响应参数选择设置对话框

【Linestyle Preferences】为 LTI 观测分析器响应线形、颜色选择设置对话框如图 5-16 所示。由图可见，LTI 观测分析器设置了响应曲线的 7 种颜色，5 种线形，9 种标识符号，可以在对话框下部的 3 个顺序框中用旁边的按钮或鼠标进行选择。颜色、线形与标识符号选择的结果，对应于系统、输入、输出、通道是否采用，还可以在对话框上部的单选按钮处进行选择。

图 5-16　分析器响应线形设置框

【例 5-6】续【例 5-1】，试用 LTI 观测器对系统同时观测单位阶跃给定响应与单位冲激给定响应曲线以及同时观测 Bode 图、Sigma 图、零极点图与 Nichols 图。

【解】

根据题目要求，用函数命令编写 MATLAB 程序 L5106.m 如下。程序中 t5101a.mdl 动态模型就是 t5101.mdl 断开闭环连接的结构图，因为绘制 Bode 图等需用系统的开环模型。

```
% MATLAB PROGRAM L5106.m
%
[a,b,c,d]=linmod2('t5101');
sys=ss(a,b,c,d);
[aa,ab,ac,ad]=linmod2('t5101a');
sysa=ss(aa,ab,ac,ad);
ltiview('step',sys)
ltiview('bode',sysa)
```

程序运行后，得到两个 LTI 观测器。可以用【Viewer Configuration】菜单命令分别设置LTI观测分析器2种响应曲线的组合以及4种频率响应的组合。也还可以用【Response Preferences】菜单命令为 LTI 观测器的响应参数进行设置以及用【Linestyle Preferences】菜单命令为 LTI 观测分析器响应线形颜色进行设置。设置完成后，即可得到同时观测单位阶跃给定响应以及单位冲激给定响应曲线，如图 5-17 所示；同时观测 Bode 图、Sigma 图、零极点图与 Nichols 图等的频域响应曲线，如图 5-18 所示。

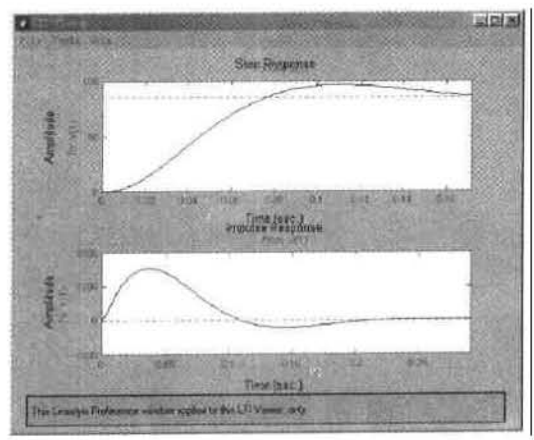


图 5-17　系统时域响应曲线

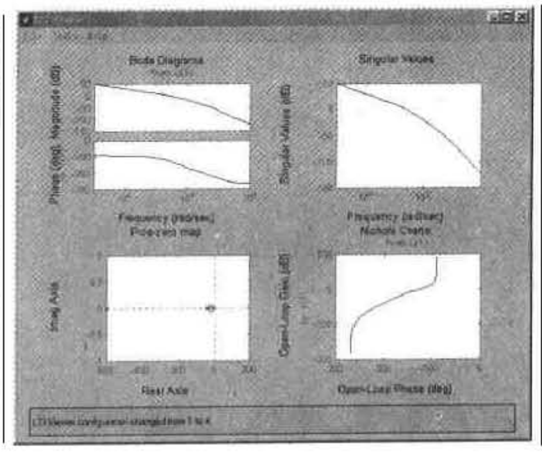


图 5-18　系统频域响应曲线

附录：

【MATLAB FUNCTION PROGRAM 4】求系统单位阶跃扰动响应性能指标(最大动态降落、最大动态降落时间、恢复时间)的函数 dist.( )

dist.( ) 函数的调用格式为：

[*detac*,*tp*,*ts*]=dist(*key*,*y*,*t*)

其中 *key* 用来选择恢复时间的%5 或 2%误差带，当 *key* =1 时表示选择%5 误差带，当 *key* =2 时表示选择 2%误差带。*y*, *t* 是对应系统单位阶跃扰动响应的函数值与时间。函

数返回的是最大动态降落 *detac*、最大动态降落时间 *tp*、恢复时间 *tv*。

```
function [detac,tp,tv]=dist(key,y,t)
% MATLAB FUNCTION PROGRAM dist.m
%
%Count detac and tp
[mp,tf]=min(y);
cs=length(t);
yss=y(cs);
detac=mp
tp=t(tf)
tm=max(t);
%Count tv
i=cs+1;n=0;
while n==0,
        i=i-1;
      if key==1,
            if i==1,
              n=1;
            elseif y(i)>=0.05,
              n=1;
            end;
      elseif key==2,
           if i==1,
              n=1;
            elseif y(i)>=0.02,
              n=1;
            end;
      end
end;
t1=t(i)
i
cs=length(t);j=cs+1;n=0;
while n==0,
      j=j-1;
         if key==1,
             if j==1,
               n=1;
             elseif y(j)<=(-0.05),
               n=1;
```

```
            end;
         elseif key==2,
            if j==1,
               n=1;
            elseif y(j)<=(-0.02),
               n=1;
            end;
       end

end;
t2=t(j)
j
      if t2<tp
        if t1>t2
          tv=t1
        end
      elseif t2>tp,
        if t2<t1,
             if t1~=0
               tv=t1
             else
               tv=t2
             end
        else
          if t1~=0
            tv=t1
          else
            tv=t2
          end
        end
      end
```

5.2 多闭环控制系统的 MATLAB 计算及仿真

5.2.1 多闭环控制系统概述

采用PI调节器的转速负反馈单闭环调速系统能够在系统稳定的前提下实现转速无静差。但是对系统的动态性能有很高要求时，还是不能令人满意。因为这种系统不能在充

分利用直流电动机过载能力的条件下获得快速响应，例如不能快速起、制动；对扰动的抑制能力也较差，例如在突加负载时系统的动态速降很大(例 5-3)等等，所以单闭环系统的应用范围受到一定限制。

在本章 5.1 节已经解释了多闭环控制系统。这种系统一环套一环，一般来说，一个闭环有一个调节器。一个调节器的输出作为下一个调节器的给定输入，最后一个调节器的输出用来调节系统的主被调量。从系统的结构来看，多个调节器是串联连接工作的，所以这样的系统又称为串级调节系统。按传统习惯，在拖动控制中叫多闭环控制系统，在过程控制中叫串级调节系统。

本节讨论的是多闭环控制系统的一个典型特例——转速、电流双闭环调速系统。该系统把转速、电流分开控制，从而设置转速、电流两个调节器，并以转速调节器的输出作为内环(电流调节器)的给定输入，于是形成转速、电流双闭环控制。这种调速系统是直流调速的一种典型基本形式，多闭环调速系统都是在此基础上发展来的。

5.2.2 多闭环控制系统的数学模型

求解单环控制系统的闭环等效传递函数在前面已经多次运算过，并要求熟练掌握，这是求解多环控制系统闭环等效传递函数的基础。然后根据多环控制系统一环套一环的特点，求解其等效传递函数的方法是从内（环）到外（环），内环的闭环等效传递函数方框作为外环里前向通道上的一个传递函数方框，再如此逐一向外环推算，直到求出多环控制系统闭环等效传递函数的最后结果。

与单闭环控制系统一样，多环控制系统数学模型的主要形式仍然是以传递函数或零极点模型为基础的系统动态结构图。

工程实际的多环控制系统仿真通常也是通过 SIMULINK 动态结构图实现的。具体过程与单闭环相同，即用 SIMULINK 提供的 linmod（）和 linmod2（）两个函数，从系统结构图模型得到一个用[A,B,C,D]表示的状态空间模型，然后对其进行各种仿真。

5.2.3 双闭环调速系统的 MATLAB 计算与仿真实例

多闭环控制系统在实际工程中应用极其广泛。为了熟悉并掌握多环系统的仿真，学会构成多环系统的 SIMULINK 动态结构图及其在图上施加给定输入信号与指定点的扰动信号的方法，训练并提高用 MATLAB 函数命令编写程序的能力，下面特举一个直流拖动双闭环 V-M 调速系统的实例。

【例 5-7】已知晶闸管-直流电机双闭环调速系统（V-M 系统）的 SIMULINK 动态结构图，如图 5-19 所示。图中直流电机 $P_{nom}=10\,\text{kW}$，$U_{nom}=220\,\text{V}$，$I_{nom}=53.5\,\text{A}$，$n_{nom}=1500\ \text{r/min}$，电枢电阻 $R_a=0.31\,\Omega$，V-M 系统主电路总电阻 $R=0.4\,\Omega$，电枢回路电磁时间常数 $T_a=0.0128\,\text{s}$，系统运动部分飞轮矩相应的机电时间常数 $T_m=0.042$ s，系统测速反馈系数 $K_t=$ 0.0067 v/rpm，系统电流反馈系数 $K_i=$ 0.072 V/A，触发整流装置的放大系数 $K_s=30$； 三相桥平均失控时间 $T_s=0.00167\,\text{s}$；电流环滤波时间常数 $T_{oi}=0.002$ s；转速环滤波时间常数 $T_{on}=0.01$ s。当忽略系统的非线性时，试分别对系统的电流内环与转速外环进行稳态与动态的计算及仿真 。

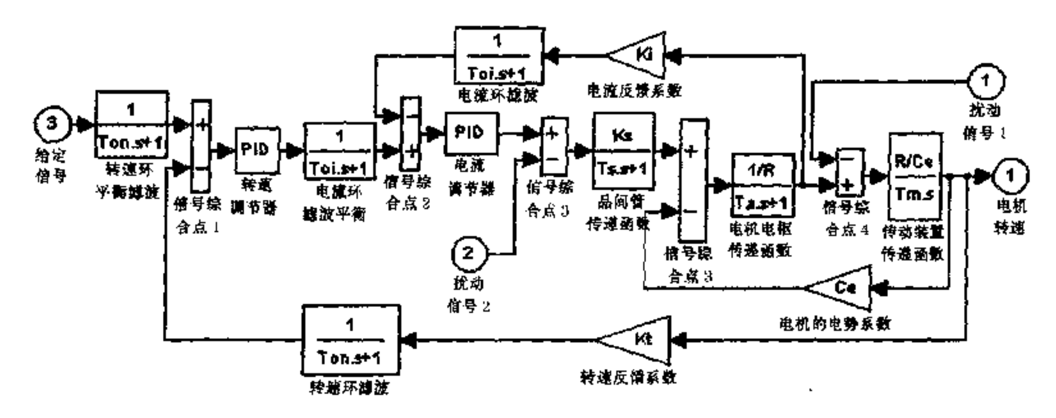


图 5-19　双闭环调速系统的 SIMULINK 动态结构图

【解】

1．说明几个问题

（1）计算额定磁通下的电机电动势转速比：

$$C_e = \frac{U_{nom} - I_{nom}R_a}{n_{nom}} = \frac{220 - 53.5 \times 0.31}{1500} = 0.1356\ \text{v} \cdot \text{min}/\text{r}$$

（2）说明电流环、转速环选型原则。双环系统的电流环一个重要作用就是保持电枢电流在动态过程中不超过允许值，突加控制作用时超调量越小越好。电流环的控制对象是双惯性型的，两个惯性时间常数之比为 $\frac{T_a}{T_{\Sigma i}} = \frac{T_a}{T_s + T_{oi}} = \frac{0.0128}{0.00167 + 0.002} = 3.49 \leqslant 10$。根据自动控制理论，一般情况下，多按典型Ⅰ型系统来设计电流环。电流环这样设计的理由是：Ⅰ型系统动态跟随性能的超调量很小，这正是电流环所要求的。转速外环是按典型Ⅱ型系统设计的。这样设计的理由是：首先，典型Ⅱ型系统具有Ⅱ型无差度。其次，典型Ⅱ型系统的抗扰动性能比Ⅰ型系统好得多。

（3）还须说明，根据线性系统的叠加原理，结构图上虽然同时绘制有多个信号作用，但在讨论其中一个信号的作用时，其他所有的信号可以认为不存在，并且在结构图上必须将其去除，只留下讨论的一个。这样的仿真 图形才是最明确清楚的。

（4）为保证电流调节器与转速调节器中的运算放大器工作在线性特性段以及保护调速系统的各个元件、部件与装置不致损坏，在电流调节器与转速调节器的输出端都设置了限幅装置。在系统结构图模型中则以饱和非线性特性来仿真，使用的为缺省设置（即上下限为+0.5 与－0.5）。

2．电流环的 MATLAB 计算及仿真

（1）电流内环动态结构图的若干考虑

由于电流环是按典型Ⅰ型系统来设计的，根据自动控制理论，电流调节器应是一个比例积分调节器。

电流环是双环系统的一个内环。在图 5-19 中，从转速调节器 ASR 的输出信号开始算起（即电流调节器的输入信号是 ASR 的输出电压），到电机电枢传递函数方框的电枢电流 $I_d(s)$ 输出信号为止，这就是电流负反馈闭环控制(图 5-20)。

采用工程上惯用的做法，已将可控硅及其触发装置近似为一阶惯性环节。

因为电流检测信号中常含有交流分量，故而须添加低通滤波措施，其滤波时间常数按基本滤平波头来进行选择。三相桥式电路每个拨头时间为 3.33 ms,为了基本滤平波头，应有$(1\sim 2)T_{oi}=3.33\text{ms}$ ，所以取 $T_{oi}=0.002\text{ s}$。由于反馈滤波同时也给反馈信号带来延迟，为平衡这一延滞作用，在给定信号通道也添加一个与反馈滤波相同时间常数的惯性环节，以使给定信号与反馈信号经过同样的延滞，这样，两者在时间上便得到了恰当的配合，同时也给设计带来很大的方便。

另外，按照把电流环设计成 I 型系统的要求，根据自动控制理论，选择电流调节器（比例积分调节器）的传递函数为：

$$W_{ACR}(s)=K_{pi}\frac{T_i s+1}{T_i s}=\frac{0.0128s+1}{0.04s},K_{pi}=0.32,T_i=0.0128$$

（2）电流内环的 SIMULINK 动态结构图

图 5-20 中给出了电流环的实际参数，也画出了仿真时给定信号与两个典型扰动信号作用点的位置。扰动信号 1 指代诸如电动机里的各种参数变化引起的扰动；扰动信号 2 指代诸如电网电压的波动变化引起的扰动以及晶闸管整流与移相触发装置参数变化所引起的扰动等。 图中未考虑反电动势的动态作用，因为反电动势信号不在环内。另外，图中在电流调节器输出端已设置了限幅装置。

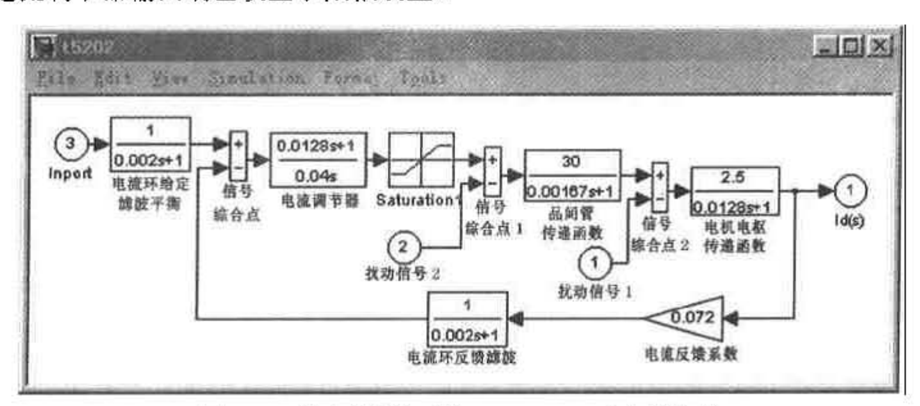


图 5-20　带参数电流环的 SIMULINK 动态结构图

（3）电流环阶跃响应的 MATLAB 计算及仿真

图 5-20 对应着 SIMULINK 动态结构图 t5202.mdl 文件。为了验证闭环系统的开环放大倍数对系统阶跃响应的影响，还将晶闸管整流与移相触发装置的放大倍数改为 $K_s=20$，其他参数不变，构成动态结构图 t5202a.mdl。

根据要求，用 linmod（）与 step（） 函数命令编写的 MATLAB 程序 L5201a.m 如下。

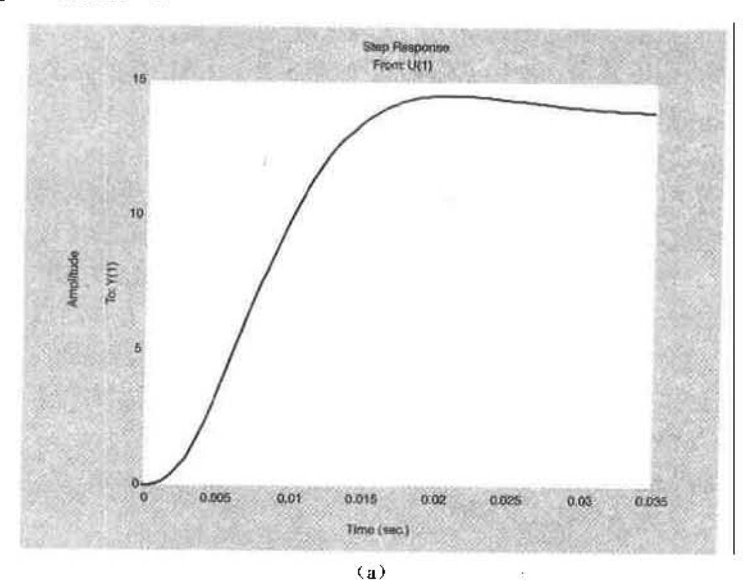
```
% MATLAB PROGRAM L5201a.m
%
[a1,b1,c1,d1]=linmod('t5202');s1=ss(a1,b1,c1,d1);
figure(1); step(s1);hold on
[a2,b2,c2,d2]=linmod('t5202a');s2=ss(a2,b2,c2,d2);
figure(2);step(s2)
[y,t]=step(s1);[mp,tf]=max(y);cs=length(t);
```

```
yss=y(cs);sigma=100*(mp-yss)/yss
tp=t(tf)
```

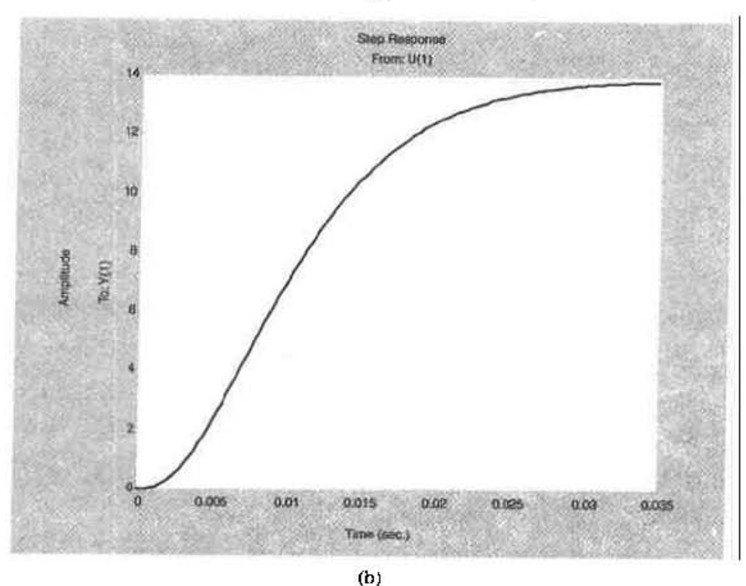
运行该程序可得模型 t5202.mdl 与 t5202a.mdl 的单位阶跃响应曲线，如图 5-21 所示，并对于图 5-21（a）求出性能指标：

超调量：*sigma*= 4.4403 %

峰值时间：*tp* = 0.0209 s



（a）



(b)

图 5-21 电流环阶跃响应 SIMULINK 曲线

图 5-21（a）是 $K_s = 30$ 时的系统单位阶跃响应，响应曲线略有超调 4.4403%，系统曲线迅速上升，峰值时间（0.0209s）非常短，电流立即下降至恒定并在 0.04s 内响应即结束，这样的阶跃响应是很理想的。这正好印证了按典型 I 型系统设计电流环其超调量

会很小的事实。电流环图 5-21（b）是 $K_x=20$ 时的系统单位阶跃响应(如果稍微减小电流环开环放大倍数的其他因子，使之为原开环放大倍数的约 0.7 倍亦可)，则阶跃响应曲线单调上升，完全无超调，并且在 0.04s 内响应亦即结束。这样的电流环阶跃响应也是很理想的。对于电流环，比较此二者，电流稍微超调的可取，因为这有利于电机的加速起动，电机又不受什么影响。

（4）电流环抗扰动响应过程的 MATLAB 计算及仿真

对绝大多数机器设备，控制系统抗扰动性能指标是至关重要的，它比起系统跟随动性能指标更为人们所关注。对于电流环，正如图 5-20 中所给出的两个作用点之扰动信号，分别指代了工程实际的一些扰动因素。

① 绘制单位阶跃扰动响应曲线并计算其性能指标

在图 5-20 中两个扰动信号作用点分别施加单位阶跃信号，绘制其扰动响应曲线，并求其最大动态降落(其值是相对系统阶跃响应稳态输出 13.5 而言的)与最大动态降落的时间。为进行仿真，在作用点 1 与 2 施加扰动信号时分别构成 MATLAB 里的动态结构图 **t5202b.mdl** 与 **t5202c.mdl**（这两个结构图中的数据同图 5-20，只是扰动信号作用点分别为 1 与 2）。并注意文件存放的路径，还须注意动态结构图中单位阶跃扰动信号的极性为负。

根据以上要求，用 linmod ( ) 与 step ( ) 函数命令并调用本书作者开发的函数 dist ( ) ，给出 如下 MATLAB 程序 L5201b.m。

```
%MATLAB PROGRAM L5201b.m
%
[a1,b1,c1,d1]=linmod('t5202b');s1=ss(a1,b1,c1,d1);
figure(1);step(s1);hold on
[a2,b2,c2,d2]=linmod('t5202c');s2=ss(a2,b2,c2,d2);
figure(2);step(s2)
[y1,t1]=step(s1); [detac,tp,tv]=dist(1,y1,t1);
[detac0,tp0,tv0]=dist(2,y1,t1);
t=[0:0.01:0.2];[y2,t2]=step(s2,t);
[detac1,tp1,tv1]=dist(1,y2,t2);[detac2,tp2,tv2]=dist(2,y1,t1);
```

程序执行后,可得图 5-22(a) 所示电流环在扰动作用点 1 施加单位阶跃信号的扰动响应曲线，并有性能指标：

最大动态降落：***detac*** = -0.9061

最大动态降落时间：***tp*** = 0.0092 s

基准值 5%范围的恢复时间：***tv*** = 0.0459 s

基准值 2%范围的恢复时间：***tv*** = 0.0580 s

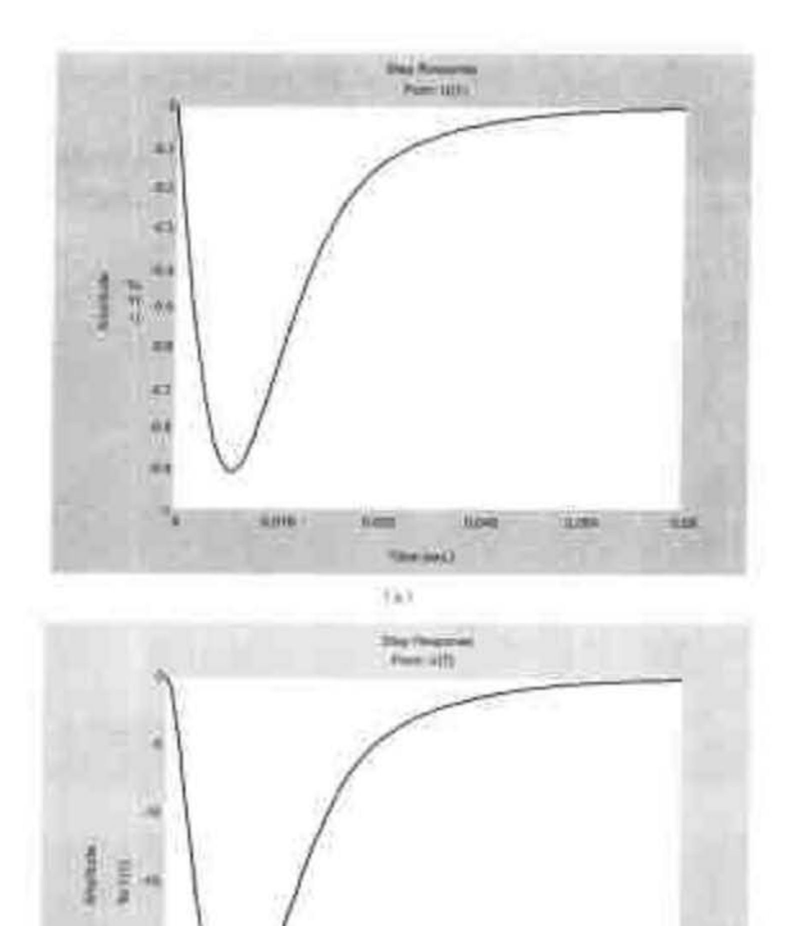
稍微更改程序，执行后可得图 5-22(b) 所示电流环在扰动作用点 2 施加单位阶跃信号的扰动响应曲线，并有性能指标：

最大动态降落：***detac*** = -26.3855

最大动态降落时间：***tp*** = 0.0106 s

基准值 5%范围的恢复时间：***tv*** = 0.0900 s

基准值 2%范围的恢复时间：***tv*** = 0.1000 s



(a)

(b)

图 5-22　电流环单位阶跃信号扰动响应曲线

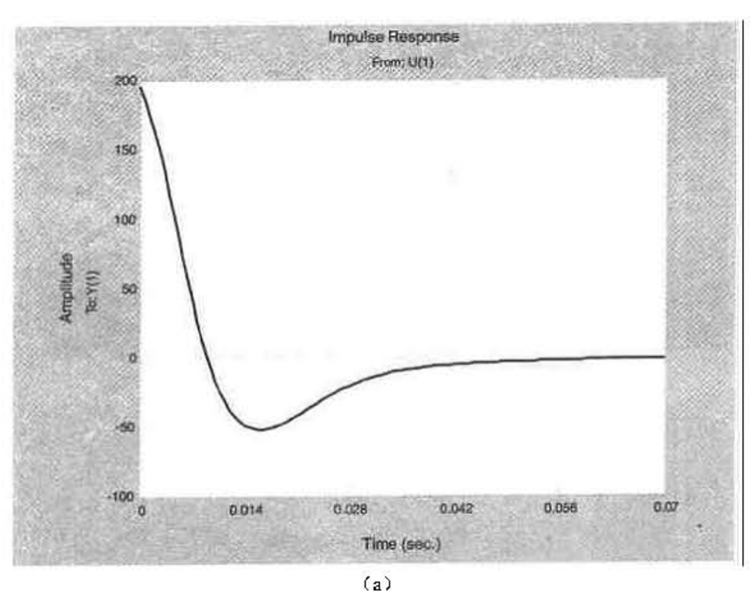
②求单位冲激信号扰动的响应曲线

在图 5-20 中两个扰动信号作用点分别施加单位冲激信号，绘制其扰动响应曲线。根据以上要求，用 linmod（）与 impulse（）函数命令编写求其扰动响应的 MATLAB 程序 L5201c.m 如下：

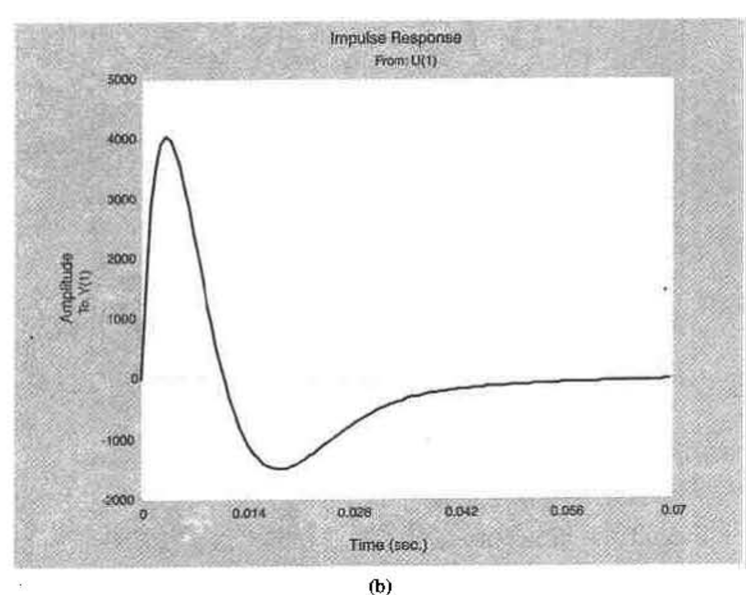
```
%MATLAB PROGRAM L5201c.m
%
[a1,b1,c1,d1]=linmod('t5202b');s1=ss(a1,b1,c1,d1);
figure(1);impulse(s1);hold on
[a2,b2,c2,d2]=linmod('t5202c');s2=ss(a2,b2,c2,d2);
figure(2);impulse(s2)
```

程序执行后,可得电流环在扰动作用点 1 与扰动作用点 2 施加单位冲激信号时的扰动响应曲线，如图 5-23(a)与图 5-23 (b)所示。

电流环冲激扰动响应过程呈单调衰减（图 5-23（a））或者呈一次衰减振荡(图 5-23(b))在 0.06 秒内就结束了，抗扰动时间非常短，系统几乎察觉不出扰动的作用。



（a）



(b)

图 5-23 电流环单位冲激信号扰动响应曲线

（5）电流环频域分析的 MATLAB 计算及仿真

根据自动控制原理，频域分析的特点是运用闭环系统的开环频率特性曲线来分析闭环系统的响应及其性能。频域分析的主要内容是画 Bode 图与计算频域性能指标。

电流闭环系统的开环结构图见图 5-24，对应着 SIMULINK 动态结构图 t5206.mdl。

根据题目要求，调用函数命令 bode （）与 margin（）的 MATLAB 指令段如下：

```
[a,b,c,d]=linmod2('t5206');
s1=ss(a,b,c,d);
margin(s1)
```

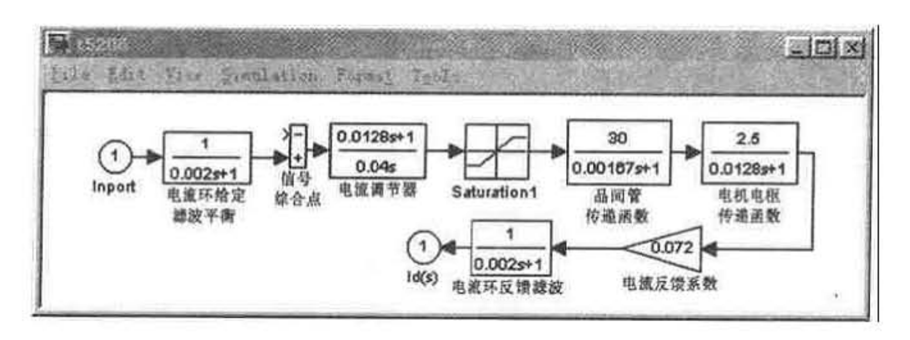


图 5-24　电流闭环系统的开环结构图

执行语句后,可得电流环的 Bode 图如图 5-25 所示，在图上就附有经计算得出的电流环的频域性能指标：

模稳定裕度	G_m=10.878 dB；	$-\pi$ 穿越频率	ω_{cg}= 306.0　s^{-1}；
相稳定裕度	P_m=50.302 deg；	剪切频率	ω_{cp}= 124.46　s^{-1}

对于图 5-24 的 SIMULINK 动态结构图，如果不考虑电流环给定平衡滤波环节的惯性作用，算出电流环的频域性能指标为：

模稳定裕度	G_m =18.3 dB；	$-\pi$ 穿越频率	ω_{cg} = 546.9　s^{-1}；
相稳定裕度	P_m=63.4 deg；	剪切频率	ω_{cp} = 127.7　s^{-1}

实际工程上，一般要求模稳定裕度 $G_m \geqslant 6$ dB，相稳定裕度 $P_m \geqslant 40$ deg。可见电流环有足够的稳定裕量，其频域性能是很好的。

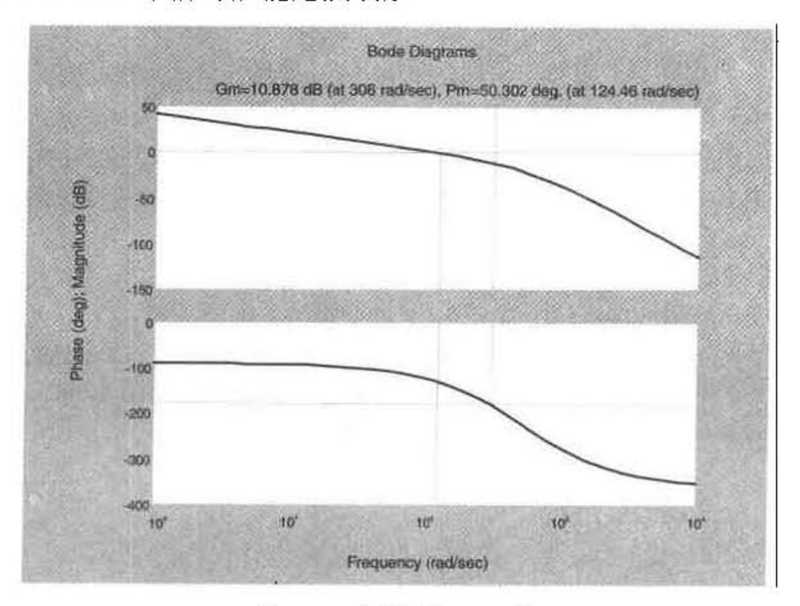


图 5-25　电流环的 Bode 图

3．转速环的 MATLAB 计算及仿真

(1) 转速环动态结构图的若干考虑

根据自动控制理论，最好转速外环按典型Ⅱ型系统设计，这样速度调节器也是一个比例积分调节器。按实际系统的构成，电流环在里面，故转速环就是一个双环系统。

为了将转速环校正成典型Ⅱ型系统，根据自动控制理论，选择转速调节器即比例积分调节器的传递函数为：

$$W_{ASR}(s)=K_{pn}\frac{T_n s+1}{T_n s}=\frac{0.0867s+1}{0.0164s},\ \ K_{pn}=5.3,\ T_n=0.0867\,\mathrm{s}$$

与电流环添加低通滤波措施一样，在转速环反馈通道与给定信号通道都添加了滤波惯性环节，其传递函数为 $\dfrac{1}{T_{on}s+1}$，式中 $T_{on}=0.01\ \ \mathrm{s}$，是根据所用测速发电机纹波情况选择的。

(2) 转速环的 SIMULINK 动态结构图（图 5-26）

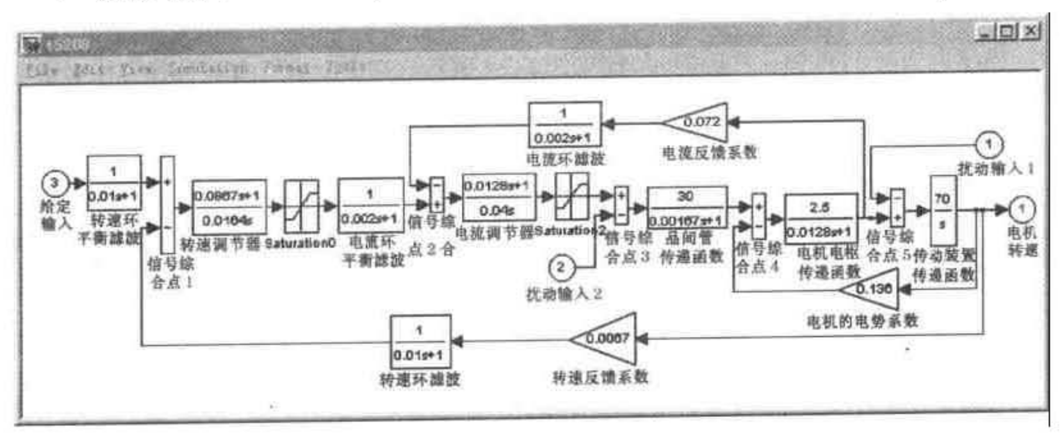


图 5-26　带参数双闭环系统的 SIMULINK 动态结构图

图 5-26 中，有电动机的电势内反馈。图中最左端是给定输入，最右端是系统输出量，即被调量—电机转速。为了讨论系统的动态扰动性能，与电流环一样，也设置了两个扰动信号作用点。扰动信号 1 指代诸如负载变化引起的扰动；扰动信号 2 仍然指代诸如电网电压的波动变化引起的扰动以及晶闸管整流与移相触发装置参数变化所引起的扰动等。

（3）转速环阶跃响应的 MATLAB 计算及仿真

根据以上要求，用 linmod ( ) 与 step ( ) 函数命令并调用函数 pref ( ) 编写求其阶跃响应与性能指标的MATLAB 程序L5201d 如下。程序中动态结构图 t5208.mdl 即是图 5-26，结构图模型 t5208b.mdl 中，$K_{pn}=5\times5.3$，K_s=100，其他参数同图 5-26。

```
% MATLAB PROGRAM L5201d.m
%
[a1,b1,c1,d1]=linmod2('t5208');
s1=ss(a1,b1,c1,d1);step(s1)
[y1,t1]=step(s1);
```

```
[a2,b2,c2,d2]=linmod2('t5208b');
s2=ss(a2,b2,c2,d2);step(s2)
[y2,t2]=step(s2);
[sigma1,tp1,ts1]=perf(1,y1,t1);
[sigma2,tp2,ts2]=perf(1,y2,t2);
```

程序执行后,可得转速环的单位阶跃响应曲线如图 5-27（a）所示。

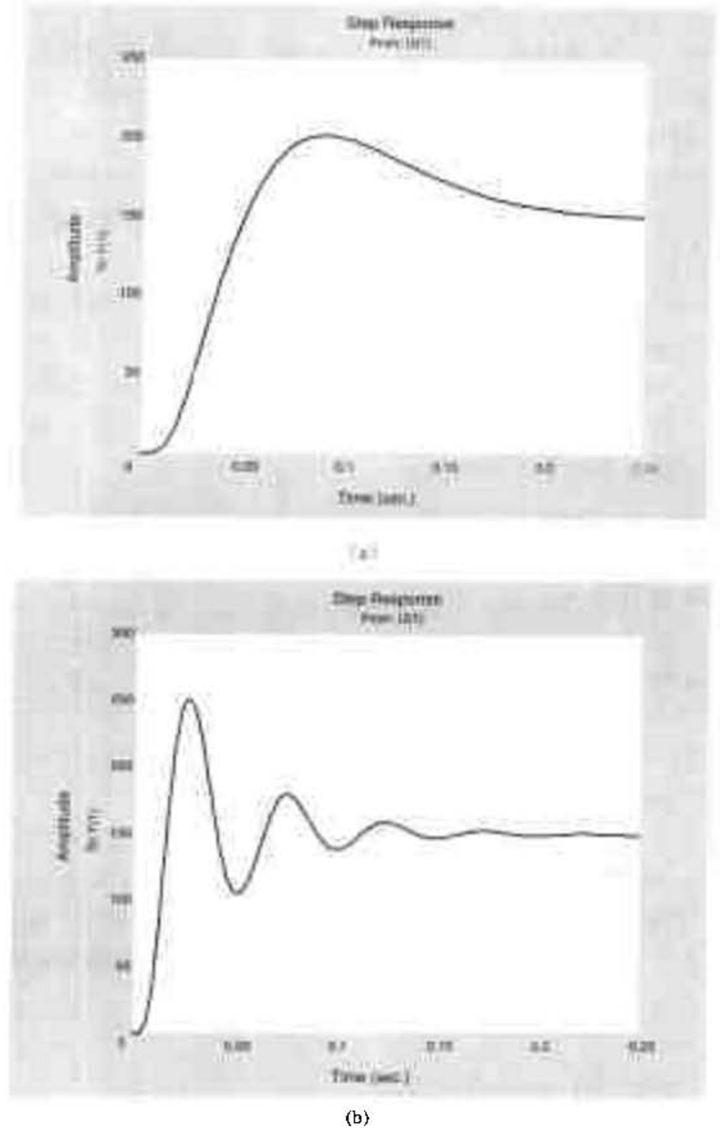


图 5-27　转速环的单位阶跃响应曲线

当图 5-26 中参数不变而仅将 K_{p_n} 加大 5 倍，且 K_s=100 时，其单位阶跃响应仿真曲线如图 5-27（b）所示，此时响应产生了衰减振荡。这正好验证了自动控制理论的提高系统开环增益会使系统稳定性变差的结论。

程序运行后，可得到图 5-27（a）阶跃响应性能指标：

超调量：*sigma*1 = 34.1602 %

峰值时间：*tp*1 = 0.0897 s

调节时间：*ts*1 = 0.1876 s

还可求得图 5-209（b）阶跃响应性能指标：

超调量：*sigma2* = 67.7645 %

峰值时间：*tp*2 = 0.025813 s

调节时间：*ts*2 = 0.10477 s （对应于 5% 的误差带）

由计算图 5-27（a）的性能指标数据可见转速环的阶跃响应超调量σ%＜35%，峰值时间 *tp*＜0.1s，调节时间 *ts*＜0.2s。这样的系统响应已是非常理想的。

（4）转速环抗扰动过程的 MATLAB 计算及仿真

控制系统抗扰动性能指标比系统跟随动性能更为重要。对于转速环，正如图 5-26 中所给出的两个作用点之扰动信号，分别指代了工程实际的一些扰动因素。

①绘制单位阶跃信号扰动响应曲线并计算其性能指标

在图 5-26 中两个扰动信号作用点分别施加单位阶跃信号，绘制其扰动响应曲线，并求其最大动态降落与最大动态降落的时间。为进行仿真，在作用点 1 与 2 施加扰动信号时分别构成 MATLAB 里的动态结构图 t5208c.mdl 与 t5208d.mdl（这两个结构图中的数据同图 5-26，只是分别表示两个扰动信号作用点）。须注意文件存放的路径，还须注意动态结构图中单位阶跃扰动信号的极性为负。

根据以上要求，用 linmod( ) 与 step( ) 函数命令并调用函数 dist ( ) 给出 MATLAB 程序 L5201e.m 如下：

```
% MATLAB PROGRAM L5201e.m
%
[a1,b1,c1,d1]=linmod2('t5208c');
s1=ss(a1,b1,c1,d1);[y1,t1]=step(s1);step(s1)
[detac,tp,tv]=dist(1,y1,t1);[detac0,tp0,tv0]=dist(2,y1,t1);
[a2,b2,c2,d2]=linmod2('t5208d');
s2=ss(a2,b2,c2,d2);t=[0:0.01:0.5];[y2,t2]=step(s2,t);step(s2,t);
[detac1,tp1,tv1]=dist(1,y2,t2);[detac2,tp2,tv2]=dist(2,y2,t2);
```

程序执行后,可得转速环的单位阶跃扰动响应曲线，如图 5-28 所示，(a)与(b)图分别对应着扰动信号作用于 1 与 2 两个不同点。对于图 5-28 (a)，计算有阶跃扰动响应性能指标：

最大动态降落：*detac* = -1.8764

最大动态降落时间：*tp* = 0.0469 s

基准值 5%范围的恢复时间：*tv* = 0.1957 s

基准值 2%范围的恢复时间：*tv* = 0.2141 s

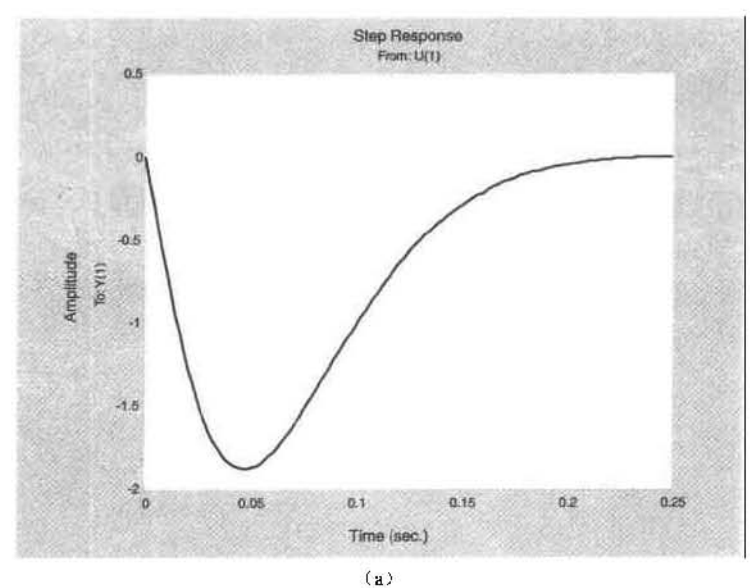
程序运行后，对于图 5-28 (b)计算有阶跃扰动响应性能指标：

最大动态降落：*detac* = -26.1314

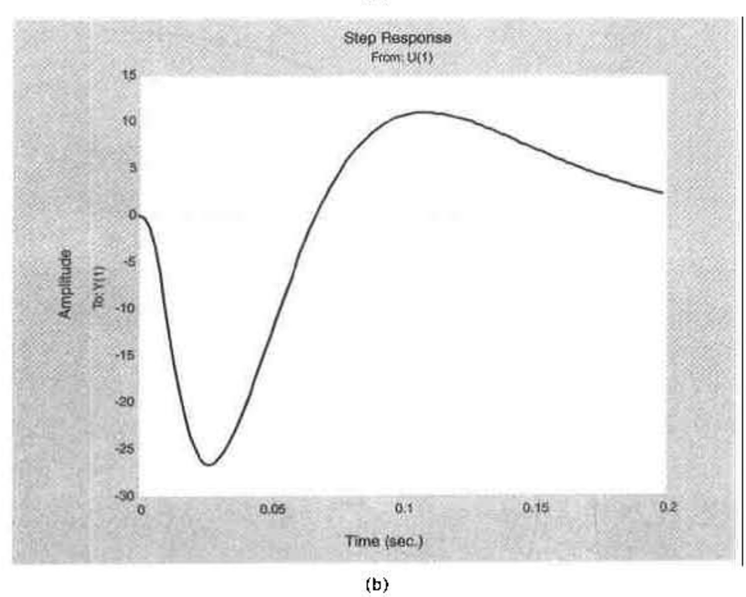
最大动态降落时间：*tp* = 0.0300 s

基准值 5%范围的恢复时间：*tv* = 0.2800 s

基准值 2%范围的恢复时间：*tv* = 0.2900 s



(a)



(b)

图 5-28 转速环的单位阶跃扰动响应曲线

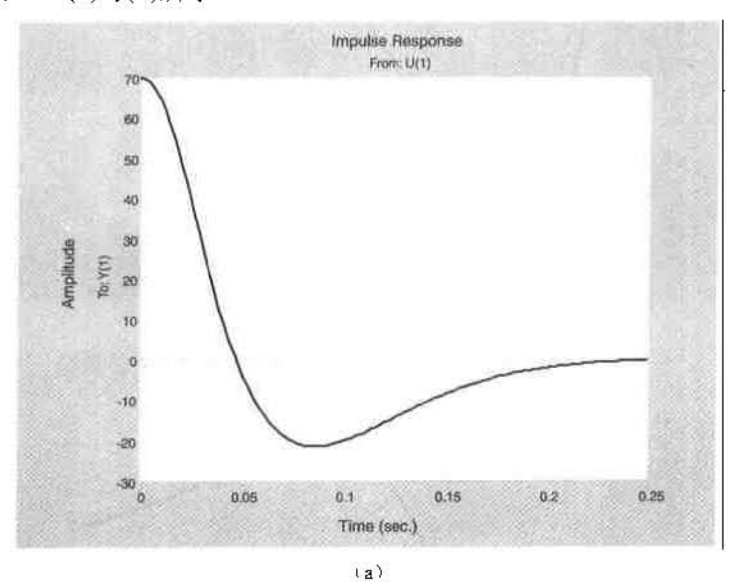
② 求单位冲激信号扰动响应曲线

在图 5-26 中两个扰动信号作用点分别施加单位冲激信号，绘制其扰动响应曲线。须注意动态结构图 t5208c.mdl 与 t5208d.mdl 中施加单位冲激扰动信号时的极性为正。根据以上要求，用 linmod ( )与 impulse ( ) 函数命令编写求其扰动响应的 MATLAB 程序 L5201f.m 如下。

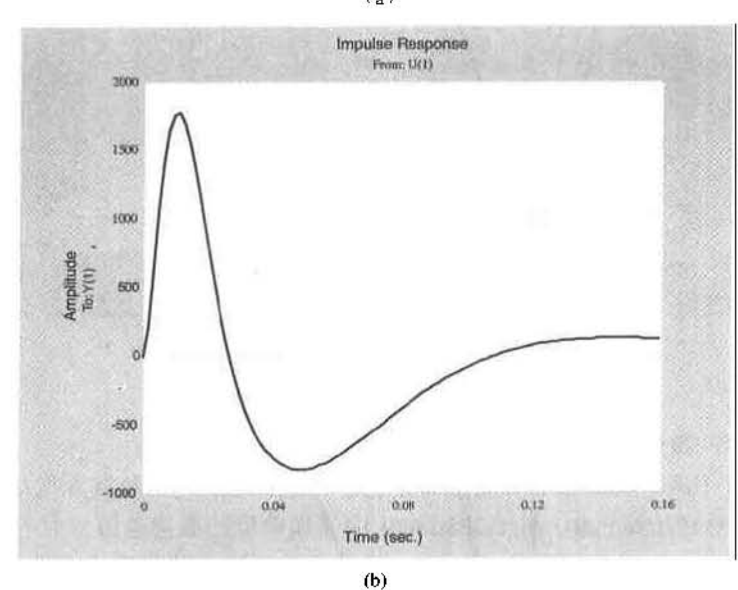
```
%MATLAB PROGRAM L5201f.m
%
[a1,b1,c1,d1]=linmod('t5208c');
figure(1);
```

```
impulse(a1,b1,c1,d1);hold on
[a2,b2,c2,d2]=linmod('t5208d');
figure(2);
impulse(a2,b2,c2,d2)
```

程序执行后,可得转速环在扰动作用点 1 与扰动作用点 2 施加单位冲激信号时的扰动响应曲线如图 5-29(a)与(b)所示。



(a)



(b)

图 5-29　转速环单位冲激信号扰动响应曲线

转速环单位冲激扰动响应过程如图 5-29（a）所示呈单调衰减，或者如图 5-29(b)所示呈一次衰减振荡在 0.2 秒内就结束了，抗扰动过程亦非常短暂，系统转速外环已经感觉到扰动，但经 0.2 秒扰动响应的过程业已结束。

（5）转速环频域分析的 MATLAB 计算及仿真

已经说明过，频域分析是运用闭环系统的开环频率特性曲线来分析闭环系统的响应与性能的。频域分析的主要内容是绘制 Bode 图与计算频域性能指标。

转速闭环系统的开环结构图，如图 5-30 所示（未画出转速环给定平衡滤波环节），此图即为 t5212.mdl 动态结构图，仿真就是对 t5212.mdl 进行操作。

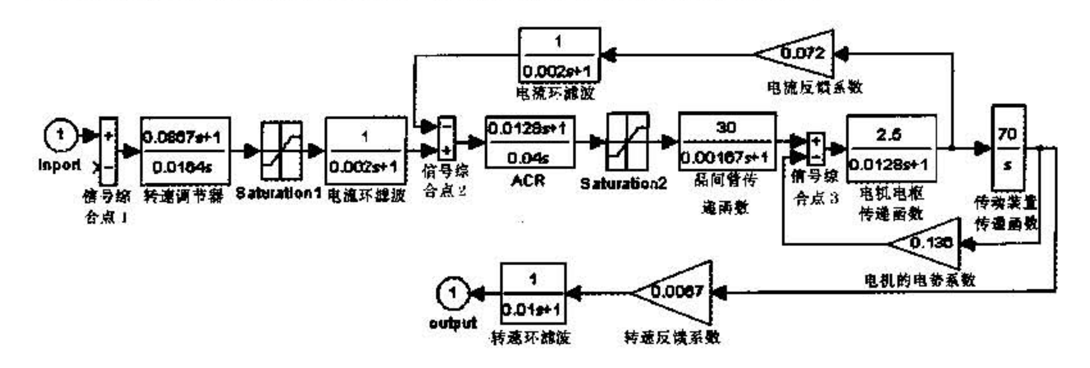


图 5-30　转速闭环系统的开环结构图

根据题目要求，用函数命令 linmod ( ) 与 margin ( ) 给出 MATLAB 指令如下。

```
[a,b,c,d]=linmod2('t5212');
sys=ss(a,b,c,d);
margin(sys)
```

语句指令执行后可得 Bode 图如图 5-31 所示，从图上可得转速环频域性能指标：

模稳定裕度 G_m=12.642 dB；　　-π 穿越频率 ω_{cg}= 99.752　s^{-1}；

相稳定裕度 P_m=43.132 deg；　剪切频率　ω_{cp} = 30.637　s^{-1}

数据显示，控制系统有较大的稳定余量。

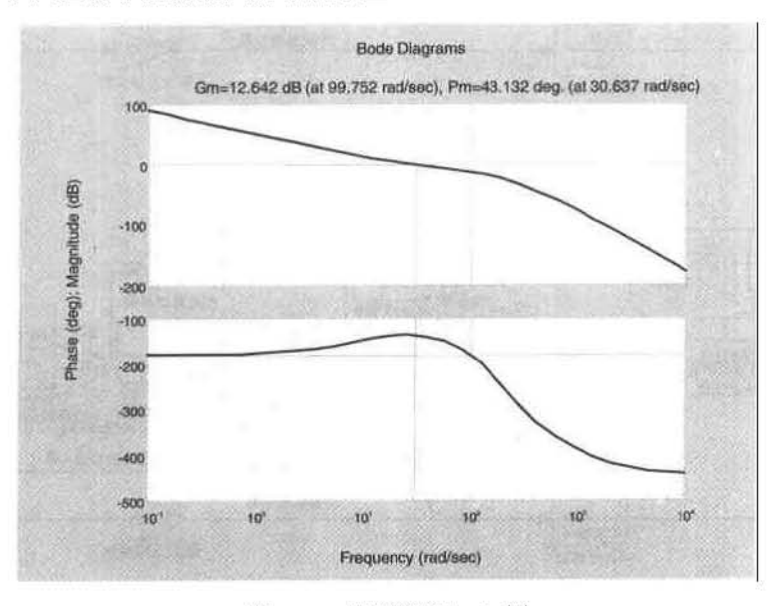


图 5-31　转速环的 Bode 图

5.2.4 三环控制系统的 MATLAB 计算与仿真

5.2.4.1 三环控制系统简述

直流双闭环调速系统广泛应用于工程实际中。但有些生产工艺要求更高，其拖动设备在双环的基础上还要增加一个外环成为三环系统。例如轧钢、造纸、纺织、印染和化纤生产中,其加工物都是带状的,并且全都卷绕成圆筒形。为使加工物不断传送，既不堆叠又不拉断，卷绕紧密，整齐，并且保证产品加工质量，在卷绕过程中，要求在加工物内建立适宜的张力并保持恒定，这就需要张力控制系统。这种张力控制系统通常都是在转速电流双闭环系统外再加一个张力环成为张力三闭环系统。

三环系统的设计通常按以下原则考虑。电流环是三环结构系统的内环，电流环的控制对象是双惯性型的，一般情况下，多按典型 I 型系统来设计电流环。对于一般无外环的转速—电流双环系统，其转速环大都按典型Ⅱ型系统设计转速环。但是对于诸如位置跟踪三环系统、张力—转速—电流的三环系统（见图 5-32），转速环外还有张力环，转速环应按典型 I 型系统进行设计。此时，转速调节器只要用比例调节器就可以了。张力环是最外环，希望动态抗扰动性能与跟随性能、静态性能都比较好，所以外环校正成为典型Ⅱ型系统是最佳选择。带参数的结构图 5-33 中，在电流调节器与张力调节器的输出端都有饱和非线性作为限幅措施，特性为缺省设置。

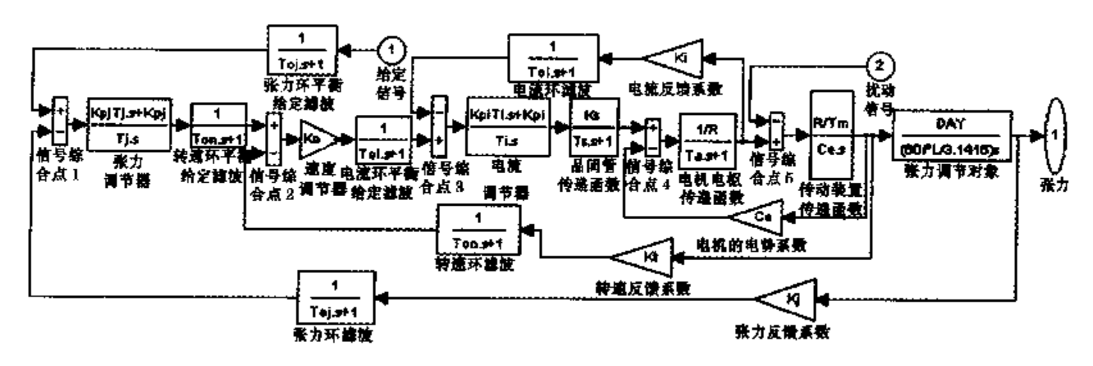


图 5-32 张力三环系统的 SIMULINK 图

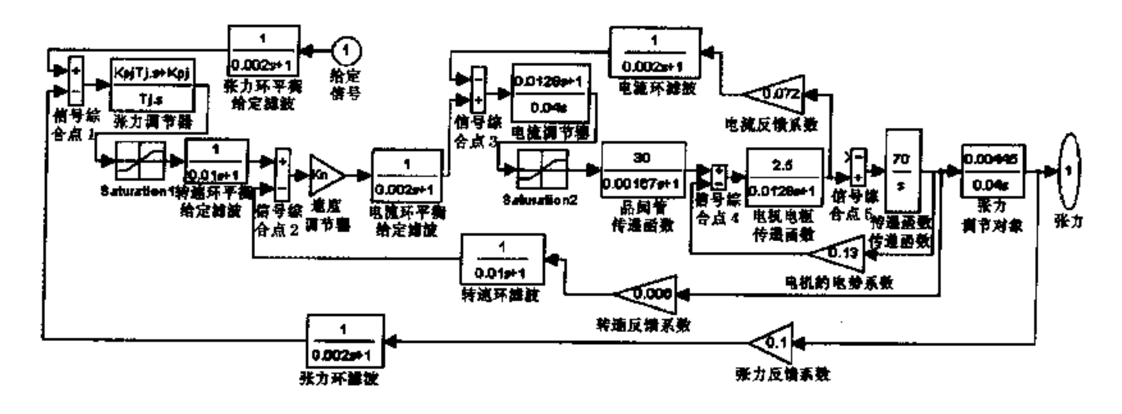


图 5-33 带参数三环系统的 SIMULINK 图

5.2.4.2 三环控制系统的MATLAB计算与仿真实例

【例 5-8】已知某设备张力三环调速系统的动态结构图如图 5-32 所示。

带参数的结构图如图 5-33 所示。图中参数：P_{nom}=10 kW; U_{nom}=220 V; I_{nom}=53.5 A; n_{nom}=1500 r/min; R=0.4 Ω; R_a = 0.31Ω; K_s = 30; T_s = 0.00167s; T_{oi} = 0.002s; T_{on} = 0.01s; K_{pi} =0.32; T_i =0.0128s; K_i = 0.072V/A; K_t = 0.0067V/(r/min); C_e = 0.136 V/(r/min); T_a = 0.0128s; T_m = 0.042s; T_{oj} = 0.002s; K_j = 0.114; D = 0.9 m; i = 10.6; $\frac{l}{AY}=0.04$。当转速调节器增益 $K_n=8$ 时，试用根轨迹设计器设计三环系统的张力调节器增益 K_{pj} 与时间常数 T_j。

【解】

已知系统结构图参数，要用 linmod()函数，提取线性模型，然后再使用 rltool()函数，给出以下 MATLAB 指令语句。t5215a.mdl 是 t5215.mdl 的开环结构图，这是绘制根轨迹图的条件。为了用根轨迹设计器来设计张力调节器，须令图中的张力调节器传递函数 $W_{ATR}(s)=1$。

```
[a,b,c,d]=linmod2('t5215a');
sys=ss(a,b,c,d);
rltool(sys)
```

指令运行后弹出根轨迹设计器如图 5-34 所示。

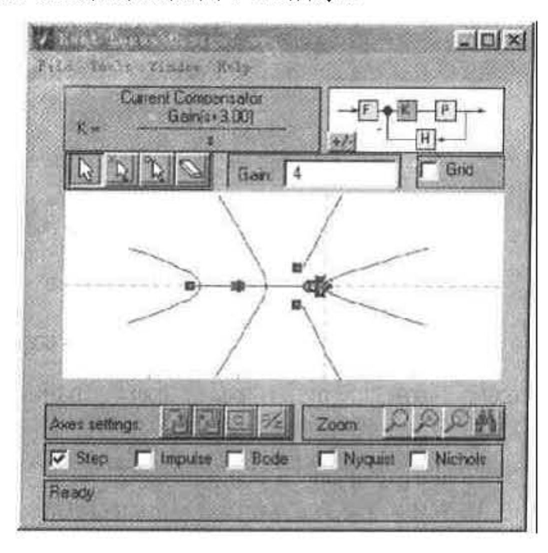


图 5-34 系统根轨迹设计器

由根轨迹设计器设计的参数（补偿器 K），得张力调节器传递函数为：

$$W_{ATR}(s)=4\frac{s+3}{s}=\frac{T_j s+1}{(T_j/K_{pj})s}=\frac{0.333s+1}{0.083s}=4.012\frac{0.333s+1}{0.333s}$$

$$\therefore\ K_{pj}=4.0112,\quad T_j=0.333$$

当转速调节器增益 $K_n=8$ 时，设计参数为张力调节器增益 K_{pj}=4.012 与时间常数 T_j=0.333s。

再给出以下程序 L5202.m 求系统单位阶跃给定响应的性能指标，程序中要调用函数 perf ( )，程序中 t5215b.mdl 是将设计参数值都代入系统的闭环结构图。

```
% MATLAB PROGRAM L5202.m
%
[a,b,c,d]=linmod2('t5215b');
sys=ss(a,b,c,d);
step(sys)
[y,t]=step(sys);
[sigma,tp,ts]=perf(2,y,t)
```

运行该程序可绘制三环系统的单位阶跃给定响应曲线如图 5-35 所示，并求出阶跃给定响应的指标：

超调量：*sigma*= 22.1489 %

峰值时间：*tp* = 0.4066 s

调节时间(2%的误差带)：*ts* = 0.9910 s

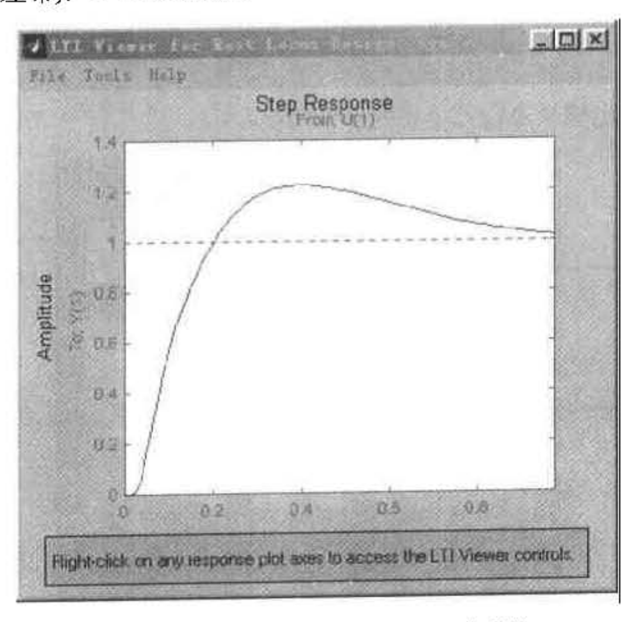


图 5-35　三环系统单位阶跃给定响应曲线

5.3　位置随动系统的 MATLAB 计算及仿真

5.3.1　位置随动系统的概念

位置随动系统是输出量对于给定输入量的跟踪系统，它实现的是执行机构对于位置

指令的准确跟踪。位置随动系统中的位置指令（给定量）和被控制量（输出量）一样，都是负载的空间位移（或者代表位移的电量），可以是角位移，也可以是直线位移。当给定量随机变化时，系统能使被控制量准确无误地跟随并复现给定量。所以位置随动系统必定是一个反馈控制系统，在实际工程中应用非常广泛。

5.3.2 位置随动系统的控制特点

与拖动控制系统相比，虽然同样是闭环控制系统，但是性能指标要求不同。对于拖动调速系统而言，给定量通常是恒值，要求系统维持输出量恒定，所以抗扰动性能指标成为主要技术指标。对于随动系统而言，给定量即位置指令是经常变化的，是一个随机变量，要求输出量准确跟随给定量的变化，因而跟随性能指标即系统输出响应的快速性、灵敏性与准确性成为它的主要性能指标。

位置随动系统需要实现位置反馈，故而系统结构上必定要包含有位置环。位置环是随动系统重要的组成部分，位置随动系统的基本特征体现在位置环上。根据位置给定信号与位置检测反馈信号综合比较的不同原理，位置随动系统分为模拟式与数字式两类。

模拟式随动系统的各种参量都是时间上连续变化的量即模拟量，这类系统的位置检测器是电位器、自整角机、旋转变压器、感应同步器等。由于模拟式位置检测装置的精度受到制造上的限制，使得模拟式随动系统的精度受到影响，其应用也受到影响。

数字式随动系统的各种参量都是数字量，这类系统的位置检测器是感应同步器、光栅、光电编码盘等。由于数字式位置检测装置的精度可以做得很高，也就使得数字式随动系统的精度是非常高的，比起模拟式随动系统来，性能要优良得多，应用也很广泛。

不论是模拟式还是数字式，位置随动系统通常是一个多闭环控制系统。闭环结构有多种不同的形式，典型的结构形式是在双闭环调速系统的基础上外加位置环，即为三环系统。也可以采用其他闭环结构控制方案，例如速度环加位置环，或者是电流环加位置环，甚至只有一个位置环，这要根据具体情况具体分析。

5.3.3 位置随动系统的分析方法与品质指标

如上所述，除随动系统的位置检测器是拖动系统里没有的之外，其他构成控制系统的元部件、装置等都与拖动系统相同，所以两者在数学模型、传递函数、动态结构图、分析方法、系统校正设计方法等方面都是完全一样的。

位置随动系统的性能指标也分动态与静态两类。动态性能指标即跟随性能指标，是阶跃响应的超调量$\sigma\%$、峰值时间t_p、调节时间t_s、振荡次数N等。静态性能指标是系统稳态精度，即稳态误差e_{ss}。这些品质指标概念的定义与拖动控制系统的相同。

5.3.4 位置随动系统 MATLAB 计算及仿真实例

【例 5-9】已知自整角机位置随动系统的动态结构图如图 5-36 所示。图中，伺服电机 S661 型，230W，110V，2.9A，2400r/min，$R_a=3.4\Omega$，电枢回路总电阻$R=5.1\Omega$，电枢回路电磁时间常数 $T_l=0.007\,\text{s}$，电机电势系数 $C_e=0.0417\,\text{V/rpm}$； 自整角机放大系数 $K_{bs}=1.25\,\text{V/(°)}$，自整角机的检测误差$e_d=0.5°$；减速器速比 i=60，减速器的放大系数

$K_g=\dfrac{6}{i}=\dfrac{6}{60}=0.1$；可控硅功率放大增益 $K_s=40$，可控硅功率放大平均失控时间 $T_s=0.00167\ \text{s}$；相敏与功放的总增益 $K_{ph}=5$，相敏与功放的时间常数 $T_{ph}=0.007\ \text{s}$；拖动系统机电时间常数 $T_m=0.9\ \text{s}$。已知串联 PID 校正器的传递函数为：

$$G_{PID}=\frac{0.2s+1}{2s+1}\cdot\frac{0.8545s+1}{0.0004681s+1}$$

试对位置随动系统进行仿真。



图 5-36　自整角机位置随动系统的动态结构图

【解】

（1）求原系统不校正时的品质指标

绘制出原系统带参数的 SIMULINK 结构图如图 5-37 所示：

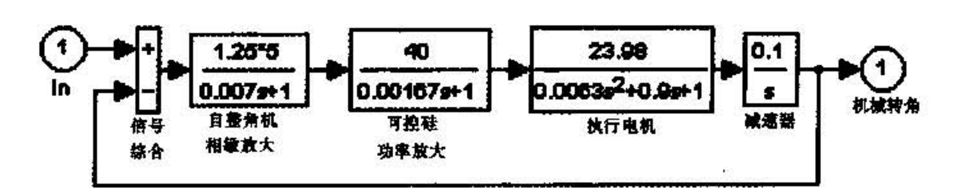


图 5-37　未校正系统的 SIMULINK 结构图

用以下 MATLAB 程序 L5301a.m 计算系统未校正时的频域性能。

```
% MATLAB PROGRAM L5301a.m
%
G1=tf(1.25,1);G2=tf(5,[0.007 1]);
G3=tf(40,[0.00167 1]);
G4=tf(23.98,[0.0063 0.9 1]);
G5=tf(0.1,[1 0]);
G=G1*G2*G3*G4*G5;
margin(G)
```

运行程序后，提示系统闭环是不稳定的，必需校正。

（2）串联 PID 校正器进行校正

根据题意，由串联 PID 校正器的结构图，给出调用函数 perf ( )、ster ( ) 计算系统校

正后频域性能、时域性能指标的 MATLAB 程序 L5301b.m 如下。

```
% MATLAB PROGRAM L5301b.m
%
G1=tf(1.25,1);G2=tf(5,[0.007 1]);G3=tf(40,[0.00167 1]);
G4=tf(23.98,[0.0063 0.9 1]);G5=tf(0.1,[1 0]);G6=tf([0.2 1],[2 1]);
G7=tf([0.8545 1],[0.0004681 1]);G=G1*G2*G3*G4*G5*G6*G7;
figure(1);[mag,phase,w]=bode(G);margin(mag,phase,w);hold on
[GM,Pm,Wcg,Wcp]=margin(mag,phase,w)
sys=feedback(G,1);figure(2);step(sys)
[y,t]=step(sys);perf(1,y,t); t=[0:0.01:30]';
[ess1]=ster(0,sys,t);[ess2]=ster(1,sys,t);
```

程序运行后，得到系统的 Bode 图（图 5-38 ）并计算出频率性能指标：

模值稳定裕度：G_m=20*log10(3.0382)=9.652 dB;

$-\pi$ 穿越频率：ω_{cg}=106.7635 s^{-1};

相角稳定裕度：P_m=39.0441 deg;

剪切频率：ω_{cp} =50.7862 s^{-1}

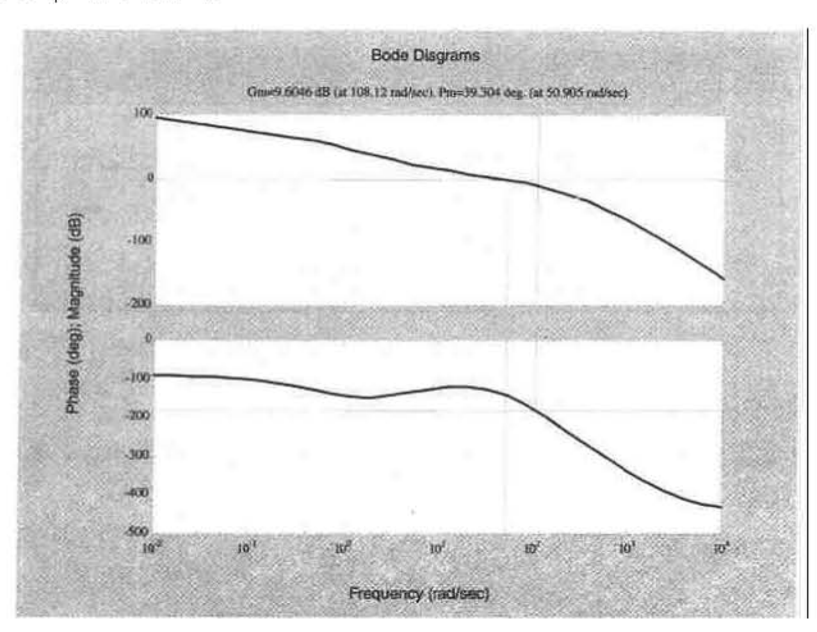


图 5-38　校正后系统的 Bode 图

程序运行后，还得到系统的单位阶跃给定响应曲线（图 5-39 ）及其性能指标：

超调量：*sigma* = 35.5791 %

峰值时间：*tp* = 0.0567 s

调节时间（5 %）：*ts* = 0.1692 s

单位阶跃给定响应的稳态误差：*ess*1 = -2.2204e-016（30 秒实际为 0 ）

单位斜坡给定响应的稳态误差：*ess*2 = 0.0017

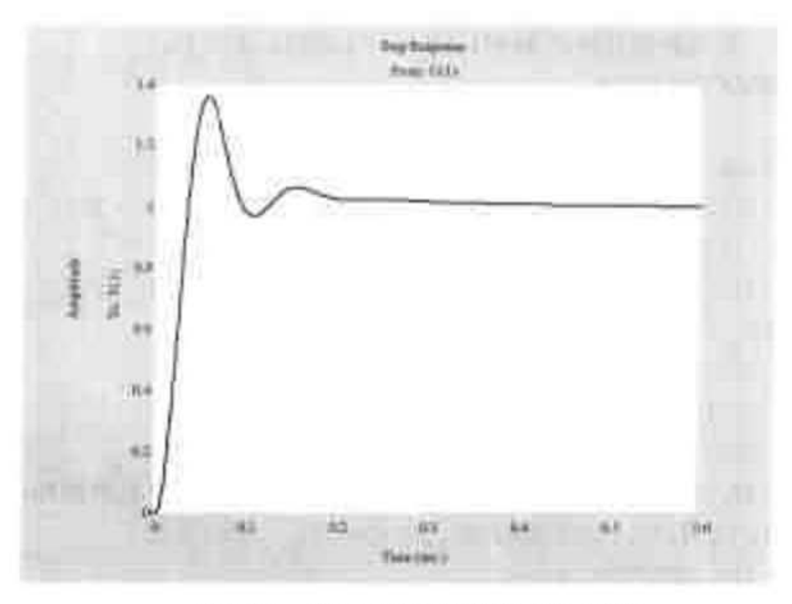


图 5-39　校正后系统的单位阶跃给定响应

【例 5-10】已知铣床光电跟踪旋转系统结构图如图 5-40 所示。



图 5-40　光电跟踪旋转系统结构图

系统各环节的传递函数分别为：

（1）从光电头偏离墨线 $\Delta\varphi$ 角，到变压器付方输出电压的幅值 $u_{\Delta\varphi}$，其间为比例环节：

$$G_1(s)=\frac{u_{\Delta\varphi}}{\Delta\varphi}\approx 7.2\,\text{V/rad}$$

（2）从交流放大到相敏整流输出其间也近似为比例环节：

$$G_2(s)=\frac{u_2}{u_{\Delta\varphi}}\approx 2$$

（3）从调节器的输入电压 $u_{in}=u_2-u_{tg}$ 到输出电压 u_{out}，此乃近似 PI 调节的传递函数：

$$G_3(s)=\frac{u_{\text{out}}(s)}{u_{\text{in}}(s)}=\frac{45(0.1s+1)}{0.5s+1}$$

（4）调节器的输出电压 u_{out} 加到永磁直流伺服电机上，电机输出角速度 Ω，直流伺

服电机传递函数为：

$$G_4(s)=\frac{\Omega(s)}{u_{out}(s)}=\frac{16}{0.08s+1}$$

（5）直流测速发电机的传递系数为：

$$G_5(s)=\frac{u_{tg}(s)}{\Omega(s)}\approx 0.007\ \text{v}\cdot\text{s/rad}$$

u_{tg} 是直流测速发电机的反馈电压。

（6）减速器的传递系数：

$$G_6(s)=\frac{0.1}{s}$$

试对双闭环旋转伺服系统进行仿真。

【解】

（1）绘制光电跟踪旋转伺服系统实际参数的 SIMULINK 动态结构图如图 5-41 所示。

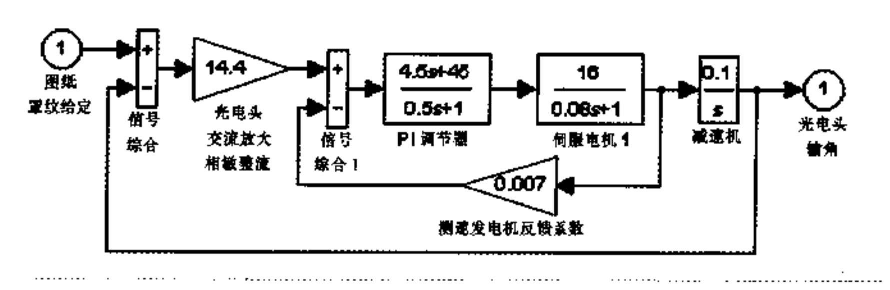


图 5-41　伺服系统的 SIMULINK 动态结构图

（2）伺服系统仿真

为进行伺服系统仿真，根据图 5-41，利用 MATLAB 的函数 linmod ( )，将其转换成系统状态空间模型即可。t5306.mdl 结构图文件就是图 5-41，t5306a.mdl 文件是系统开环结构图。以下为仿真程序 L5302a.m 与 L5402b.m。

```
%MATLAB PROGRAM L5302a.m
%
[a,b,c,d]=linmod2('t5306');sys1=ss(a,b,c,d); step(sys1)
[y,t]=step(sys1);[sgm,tp,ts]=perf(1,y,t);
sys=tf(sys1);t=[0:0.01:300]';
[ess1]=ster(0,sys,t);[ess2]=ster(1,sys,t);
```

程序运行后得到图 5-42 所示伺服系统的阶跃给定响应曲线，并计算出性能指标：

单位阶跃给定响应的超调量：*sigma* = 59.0423 %

单位阶跃给定响应的峰值时间：*tp* = 0.0633 s

单位阶跃给定响应的调节时间（5%）：*ts* = 0.273 s

单位阶跃给定响应的稳态误差：*ess*1 = -1.3323e-015（此为截断误差引起 实际为 0）

单位阶跃斜坡响应的稳态误差：*ess*2 - 0.0058

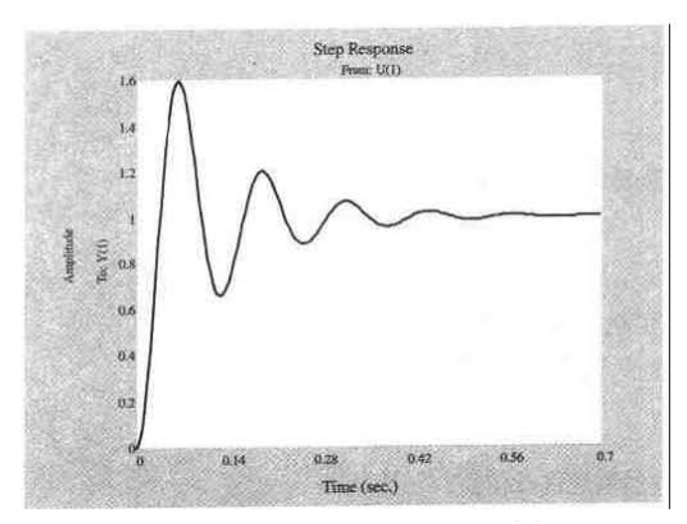


图 5-42　伺服系统的阶跃给定响应

```
%MATLAB PROGRAM L5302b.m
%
[a,b,c,d]=linmod2('t5306a');
[mag,phase,w]=bode(a,b,c,d);
margin(mag,phase,w)
[GM,Pm,Wcg,Wcp]=margin(mag,phase,w)
```

程序运行后，得到系统的 Bode 图（图 5-43），并计算出频率性能指标：

模值稳定裕度：G_m=20*log10(23.7051)=27.5 dB;

$-\pi$ 穿越频率：ω_{cg}= 250.4798 s^{-1};

相角稳定裕度：P_m=18.9122 deg;

剪切频率：ω_{cp} =. 49.3217 s^{-1}

这样的系统，就是超调量大了一些，相角稳定裕度稍小，其他性能指标还是可以的。

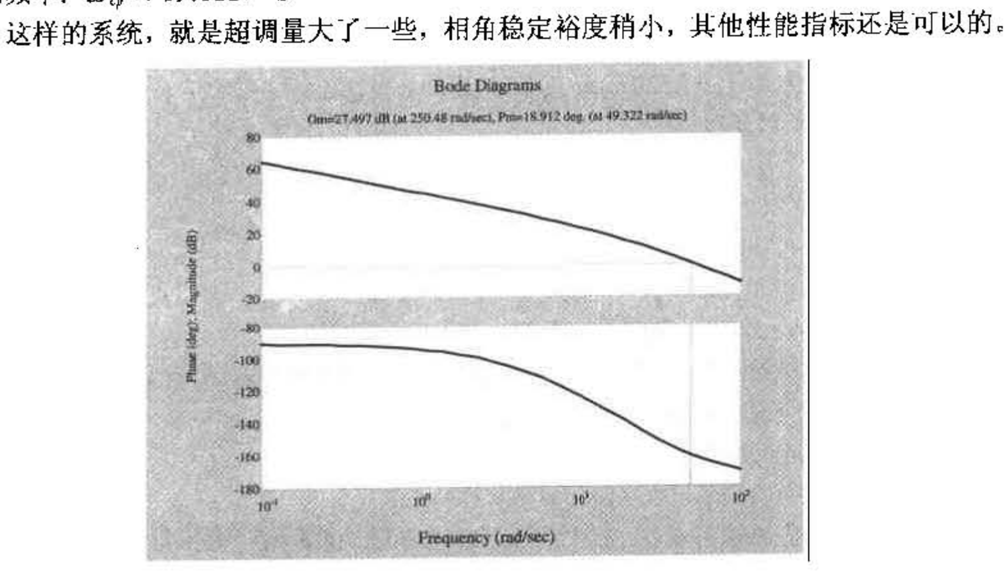


图 5-43　伺服系统的 Bode 图

【例 5-11】一个小功率角度跟踪复现系统结构图如图 5-44 所示，整定系统可调放大系数 $K_1=1$，$K_2=800$，滤波器时间常数 $T_L=0.0024$ s，伺服电机拖动系统电时间常数 $T_M=0.2$ s，串联 PD、PID 校正器传递函数为：

$$G_{PD}(s)=\frac{0.028s+1}{0.0015s+1};\quad G_{PID}(s)=\frac{(0.2s+1)(0.0435s+1)}{(0.416s+1)(0.0024s+1)};\quad H(s)=\frac{0.000452s^2}{0.0435s+1}$$

试对位置随动系统 PD、PID 串联校正与 $H(s)$ 反馈校正进行全面仿真。

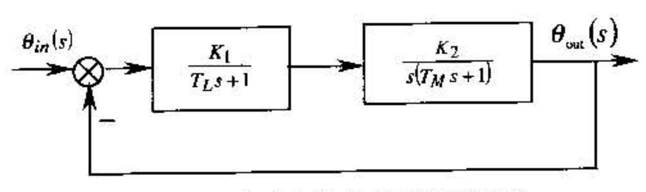


图 5-44　小功率角度复现系统结构图

【解】

（1）求原系统未校正时的阶跃给定响应

用以下 MATLAB 程序 L5303a.m 绘制系统未校正时的阶跃给定响应曲线。

```
% MATLAB PROGRAM L5303a.m
%
G1=tf(1,[0.0024 1]);G2=tf(800,[0.2 1 0]);
Gc1=G1*G2;Gcc1=feedback(Gc1,1);
step(Gcc1)
```

运行程序后，系统未校正时的阶跃给定响应如图 5-45 所示，曲线是发散振荡的，这样的系统是不能运行的，必须进行校正。

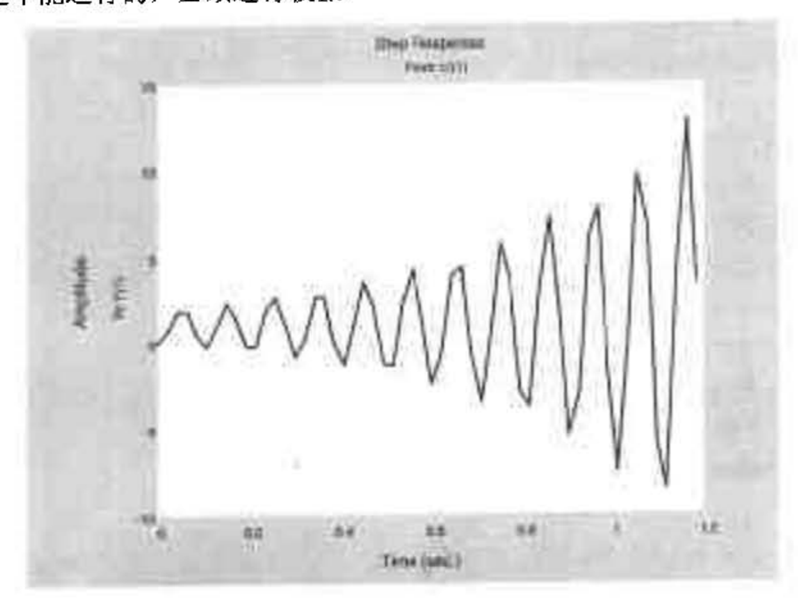


图 5-45　系统不校正时的阶跃给定响应曲线

（2）采用 PD 调节器进行串联校正

由 PD 校正系统结构图 5-46，用以下 MATLAB 程序 L5303b.m 计算校正后的阶跃给定响应与频域性能指标。

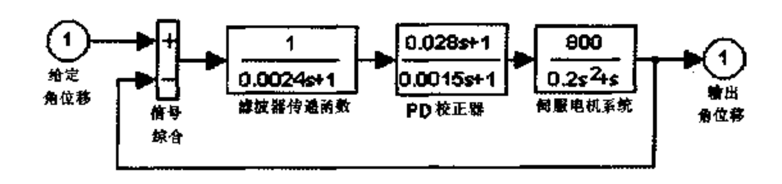


图 5-46　PD 校正系统 SIMULINK 结构图

```
% MATLAB PROGRAM L5303b.m
%
G1=tf(1,[0.0024 1]);G2=tf(800,[0.2 1 0]);
G3=tf([0.028 1],[0.0015 1]);Gc2=G1*G2*G3;
figure(1);[mag,phase,w]=bode(Gc2);
margin(mag,phase,w);hold on
[Gm,Pm,Wcg,Wcp]=margin(mag,phase,w)
sys=feedback(Gc2,1);figure(2);step(sys)
[y,t]=step(sys);perf(1,y,t); t=[0:0.1:10]';
[ess1]=ster(0,sys,t);[ess2]=ster(1,sys,t);
```

程序运行后，得到系统的 Bode 图 5-47 及系统的阶跃给定响应曲线图 5-48，并计算出频率与时域性能指标：

模值稳定裕度：G_m =20*log10(8.6279)= 18.72 dB

$-\pi$ 穿越频率：ω_{cg} = 493.4694 s^{-1}

相角稳定裕度：P_m = 49.7582 deg

剪切频率：ω_{cp}= 111.8906 s^{-1}

系统阶跃响应超调量：*sigma* = 23.7506 %

系统阶跃响应峰值时间：*tp* = 0.0249 s

系统阶跃响应调节时间：*ts* = 0.0524 s

系统单位阶跃给定响应稳态误差：*ess* = -1.7764e-015 （此为截断误差引起，实际为 0）

系统单位斜坡给定响应稳态误差：*ess* = 0.0012

（3）采用 PID 调节器进行串联校正

对于 PD 校正系统结构图，用 PID 校正器替换 PD 校正器后，用以下 MATLAB 程序 L5303c.m 计算校正后的阶跃给定响应与频域性能指标。

```
% MATLAB PROGRAM L5303c.m
%
G1=tf(1,[0.0024 1]);G2=tf(800,[0.2 1 0]);
G3=tf(conv([0.2 1],[0.0435 1]),conv([0.416 1],[0.0024 1]));
Gc2=G1*G2*G3;
```

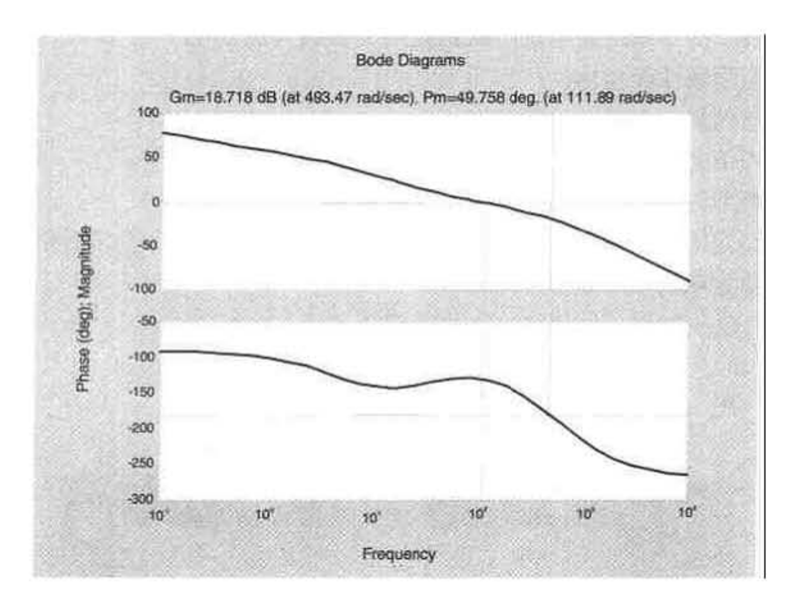


图 5-47　PD 校正后的 Bode 图

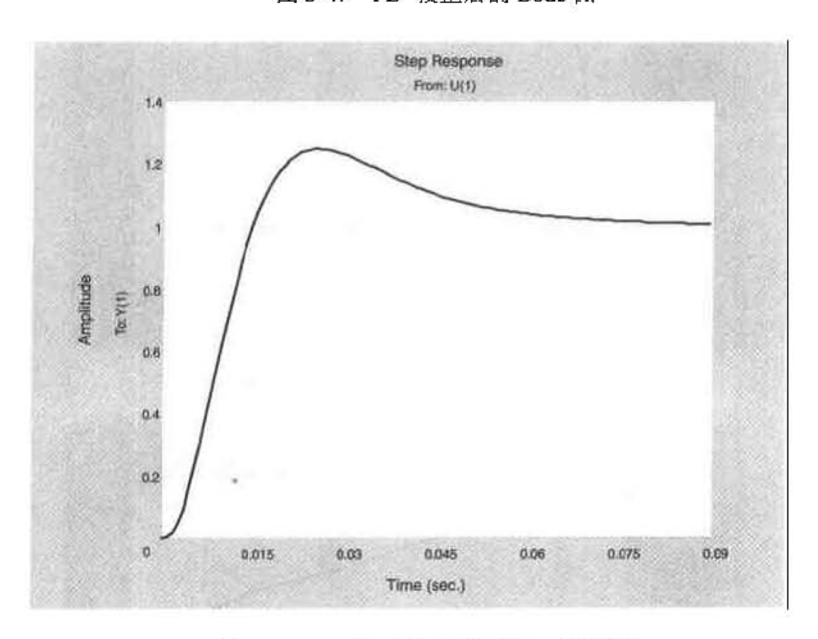


图 5-48　PD 校正后的阶跃给定响应曲线

```
figure(1);[mag,phase,w]=bode(Gc2);
margin(mag,phase,w);hold on
[Gm,Pm,Wcg,Wcp]=margin(mag,phase,w)
sys=feedback(Gc2,1);figure(2);step(sys)
[y,t]-step(sys);perf(1,y,t);
t=[0:0.1:10]';[ess1]=ster(0,sys,t);
[ess2]=ster(1,sys,t);
```

程序运行后，得到系统的单位阶跃给定响应曲线（图 5-49）与 Bode 图（图 5-50），并计算出时域与频域性能指标：

系统阶跃响应超调量：*sigma* = 20.3969 %

系统阶跃响应峰值时间：*tp* = 0.0346 s

系统阶跃响应调节时间：*ts* = 0.0779 s

系统单位阶跃给定响应稳态误差：*ess*1 = -2.2204e-016（此为截断误差引起 实际为 0）

系统单位斜坡给定响应稳态误差：*ess*2 = 0.0012

模值稳定裕度：G_m =20*log10(9.1436)= 19.222 dB

$-\pi$ 穿越频率：ω_{cg} = 395.4582 s^{-1}

相角稳定裕度：P_m = 53.498 deg

剪切频率：ω_{cp} = 83.4037 s^{-1}

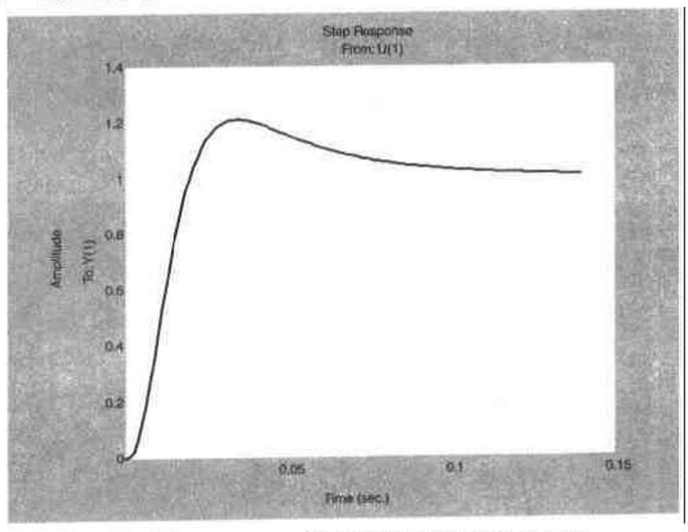


图 5-49　PID 校正后的阶跃给定响应曲线

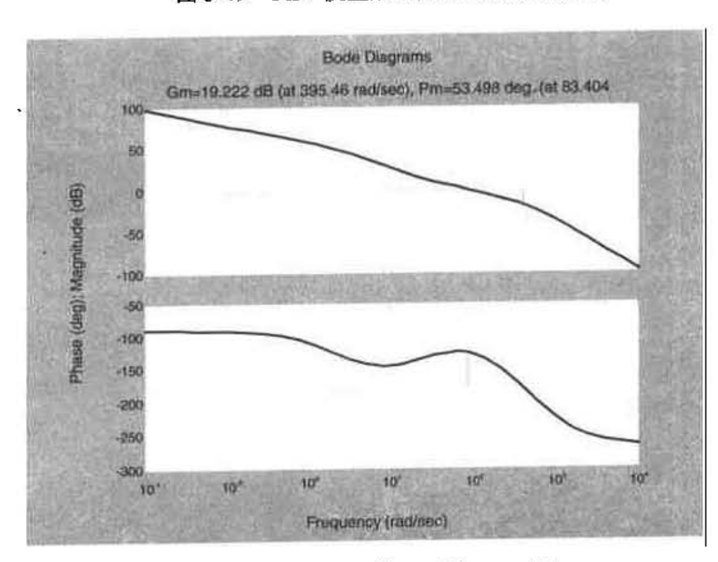


图 5-50　PID 校正后的 Bode 图

（4）采用 $H(s)$ 反馈校正器

①绘制反馈校正闭环系统的 SIMULINK 结构图如图 5-51 所示。

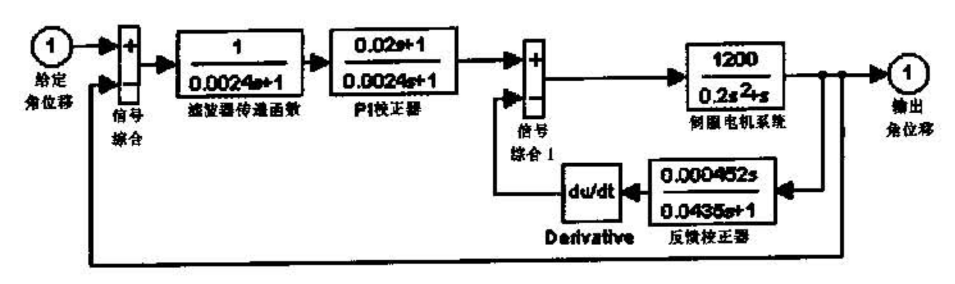


图 5-51　反馈校正闭环系统 SIMULINK 结构图

② 对反馈校正闭环系统进行阶跃给定响应仿真。由反馈校正系统结构图，用以下 MATLAB 程序 L5303d.m 计算校正后的阶跃给定响应与时域性能指标。

```
% MATLAB PROGRAM L5303d.m
%
[a,b,c,d]=linmod2('t5316');sys1=ss(a,b,c,d);step(sys1)
[y,t]=step(sys1);[sgm,tp,ts]=perf(1,y,t);
sys=tf(sys1);t=[0:0.1:10]';
[ess1]=ster(0,sys,t);
[ess2]=ster(1,sys,t);
```

程序运行后，得到系统的阶跃给定响应曲线图 5-52，并计算出时域性能指标：

系统阶跃响应超调量：*sigma* = 40.8364 %

系统阶跃响应峰值时间：*tp* = 0.0233 s

系统阶跃响应调节时间：*ts* = 0.0417 s

系统单位阶跃给定响应稳态误差：*ess*1 = 6.3505e-014 （此为截断误差引起，实际为 0）

系统单位斜坡给定响应稳态误差：*ess*2 = 8.3333e-004

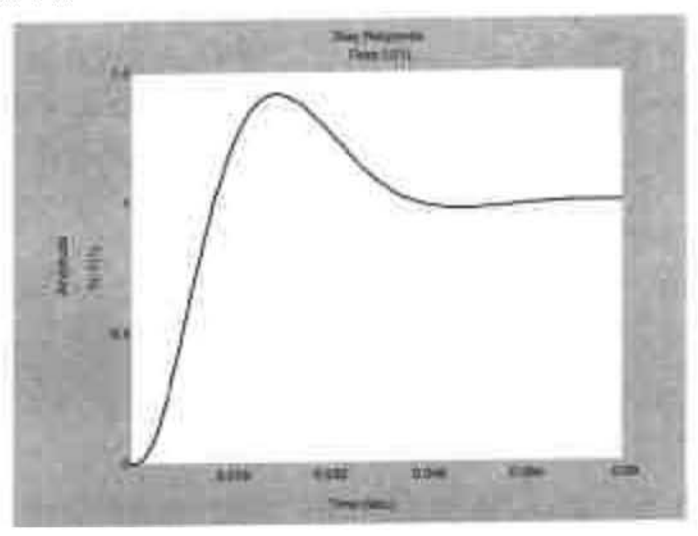


图 5-52　反馈校正后的阶跃给定响应曲线

③绘制反馈校正开环系统的 SIMULINK 结构图，如图 5-53 所示。

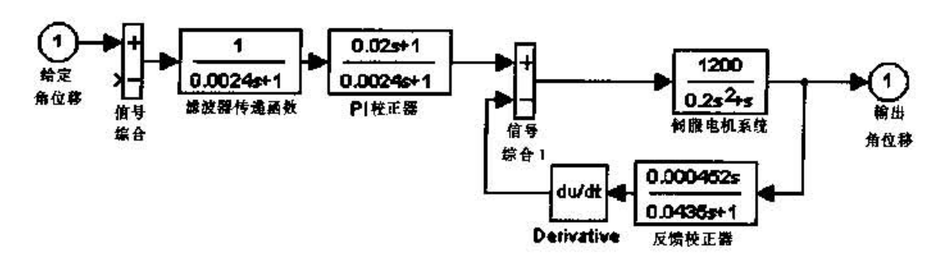


图 5-53　反馈校正开环系统 SIMULINK 结构图

由反馈校正开环系统结构图 5-53，用以下 MATLAB 程序 L5303e.m 计算校正后的频域性能指标。

```
% MATLAB PROGRAM L5303e.m
%
[a,b,c,d]=linmod2('t5318');
[mag,phase,w]=bode(a,b,c,d);
margin(mag,phase,w)
[Gm,Pm,Wcg,Wcp]=margin(mag,phase,w)
```

程序运行后，得到系统的 Bode 图（图 5-54）并计算出频域性能指标：

模值稳定裕度：G_m =20*log10(5.4668)= 14.8 dB

$-\pi$ 穿越频率：ω_{cg} = 366.9361 s^{-1}

相角稳定裕度：P_m = 37.2923 deg

剪切频率：ω_{cp} = 119.9889 s^{-1}

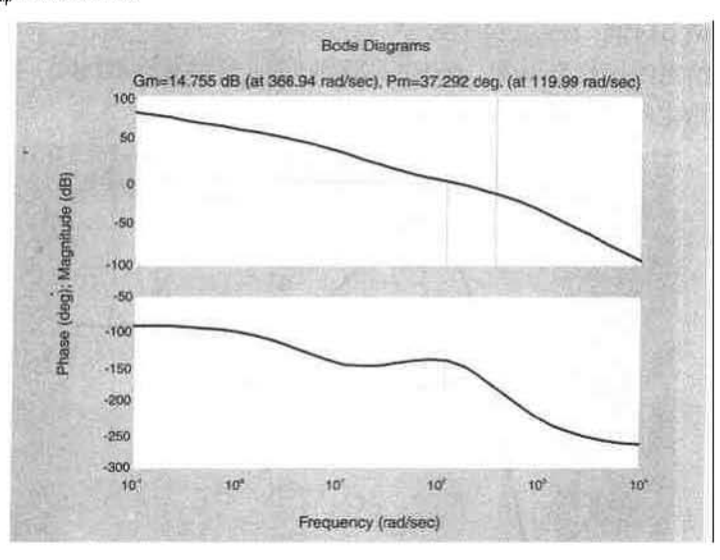


图 5-54　反馈校正开后的 Bode 图

系统经串联 PD、PID 与反馈校正后，系统的频域、时域性能指标均为优良。超调量最大的（反馈校正）限制在 40 %以下，串联校正的都在 25 %左右。串联校正的相角稳定

裕度都大于 45°，反馈校正的略低于 40°。剪切频率与-π穿越频率均在 110 s^{-1}~350 s^{-1} 之间，快速性能是很好的。实际应用时，可根据具体要求选择串联校正还是反馈校正。

【例 5-12】某一锁相位置伺服系统的结构图如图 5-55 所示。调速部分是大功率晶闸管供电的脉宽调制式调速系统（PWM），K_j、T_j 是鉴相滤波环节的放大系数与时间常数：K_j=1.11，T_j=0.0132 s。图中参数：$K_e=0.133$，$T_a=0.0035$ s，K_i=0.26，K_t=0.01，K_f=2.5，$T_m=0.116$ s，$T_1=0.0005$ s，K_1=33.3，K_2=0.5，K_3=15.05。晶闸管电机主回路总电阻：$R=2\Omega$。电流调节器 ACR、转速调节器 ASR、位置调节器 AWR 的传递函数分别为：

$$W_{ACR}(s)=K_{pi}\frac{T_i s+1}{T_i s}=\frac{2s+1}{2s},K_{pi}=1,T_i=2$$

$$W_{ASR}(s)=K_{pn}\frac{T_n s+1}{T_n s}=200\frac{\frac{1}{30}s+1}{\frac{1}{30}s},K_{pn}=200,T_n=\frac{1}{30}$$

$$W_{AWR}(s)=K_{pw}\frac{T_{w1} s+1}{T_{w2} s+1}=\frac{4.73s+118}{s+50}=2.36\frac{0.04s+1}{0.02s+1}$$

$$K_{pw}=2.36,T_{w1}=0.04,T_{w2}=0.02$$

试对位置随动系统进行全面仿真。

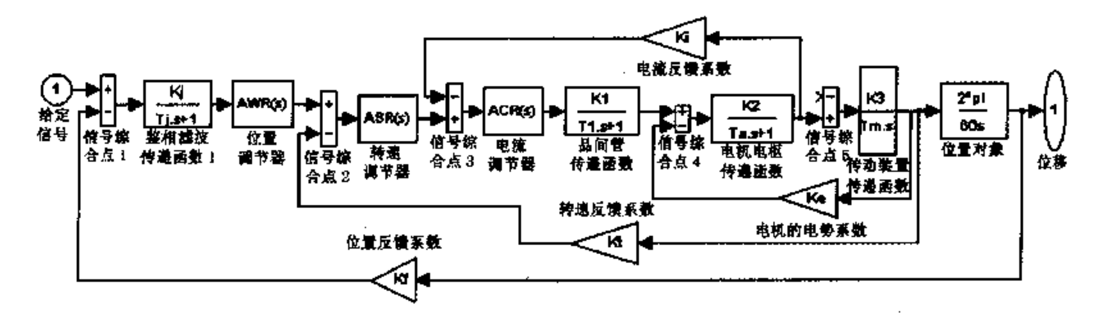


图 5-55　锁相位置伺服三环系统

【解】

（1）对位置伺服系统电流环进行仿真

①绘制电流环结构图如图 5-56 所示。

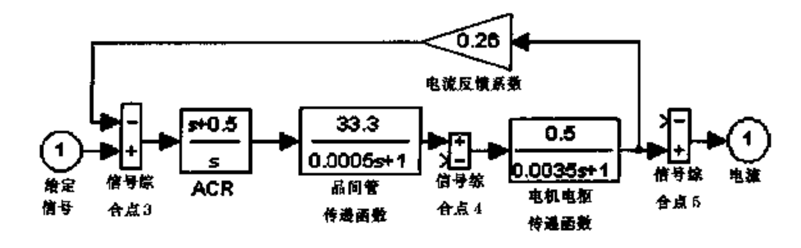


图 5-56　锁相位置伺服系统的电流环

②对位置伺服系统的电流环进行时域仿真。

对系统电流环进行时域仿真的程序段如下。

```
[a,b,c,d]=linmod2('t5321');sy=ss(a,b,c,d);
sys=tf(sy);t=[0:0.001:0.1];
step(sys,t);
```

程序段运行后得到电流环阶跃给定响应曲线如图 5-57 所示。

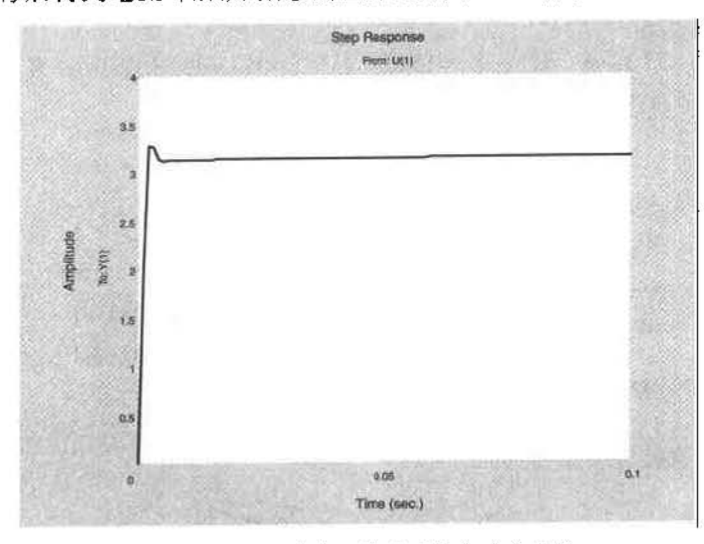


图 5-57　电流环的阶跃给定响应曲线

③ 对位置伺服系统电流环进行频域仿真

对系统电流环进行频域仿真的程序段如下。

```
[a,b,c,d]=linmod2('t5321a');
sy=ss(a,b,c,d);sys=tf(sy);margin(sys)
[Gm,Pm,Wcg,Wcp]=margin(sys)
```

仿真程序中用的 t5321a.mdl 模型就是电流环图 5-56 的开环结构图。程序段运行后得

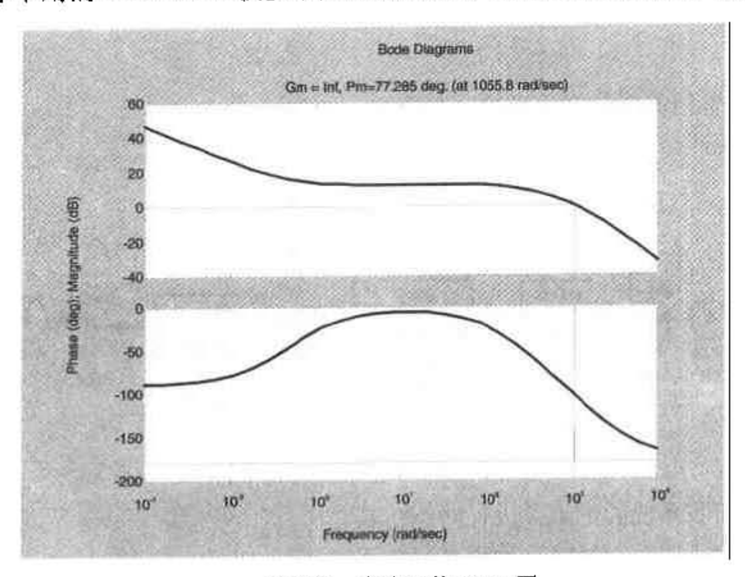


图 5-58　电流环的 Bode 图

到电流环的 Bode 图（图 5-58）与频域性能指标：

模稳定裕度 G_m=Inf；　　　　　-π 穿越频率 ω_{cg}= NaN；

相稳定裕度 P_m=77.285 deg；　　　剪切频率 ω_{cp} = 1055.8 s^{-1}

（2）对位置伺服系统转速环进行仿真

①绘制转速环结构图如图 5-59 所示。

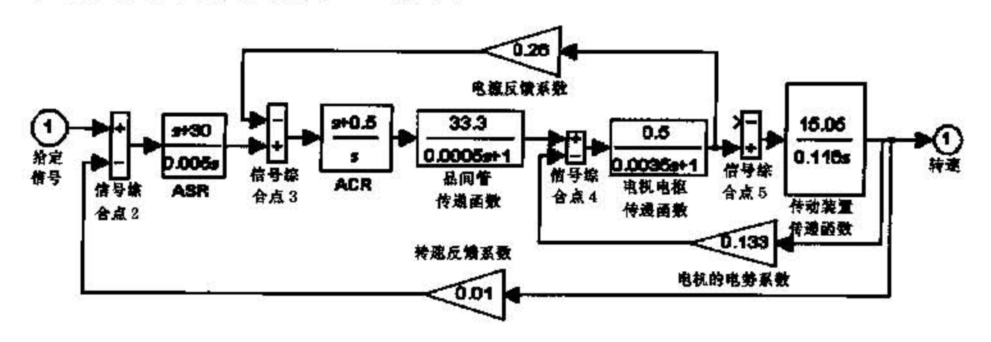


图 5-59　锁相位置伺服系统转速—电流环

②对位置伺服系统转速环进行时域仿真。对系统转速环进行时域仿真的程序段如下：

```
[a,b,c,d]=linmod2('t5324');
sy=ss(a,b,c,d);sys=tf(sy);
t=[0:0.001:0.1];step(sys,t)
[y,t]=step(sys);[sgm,tp,ts]=perf(1,y,t);
```

程序段运行后得到转速—电流环阶跃给定响应曲线（如图 5-60 所示）及其时域响应性能指标：

超调量：*Sigma* = 22.6547 %

峰值时间：*tp* = 0.0032 s

调节时间：*ts* = 0.0043 s

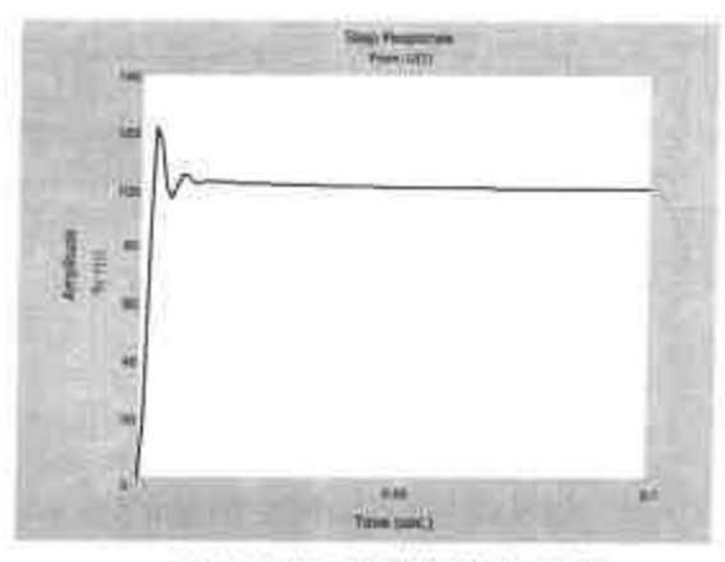


图 5-60　转速-电流环的阶跃给定响应曲线

③对位置伺服系统转速环进行频域仿真。仿真程序中用的 t5324a.mdl 模型就是转速环图 5-59 的开环结构图。经仿真可得到转速—电流环的 Bode 图（图 5-61）及其频域性能指标：

模稳定裕度 $G_m = 8.8013$ dB；　-π 穿越频率 $\omega_{cg} = 1725.7\ \ \mathrm{s}^{-1}$；

相稳定裕度 $P_m = 49.815$ deg；　剪切频率 $\omega_{cp} = 817.84\ \ \mathrm{s}^{-1}$

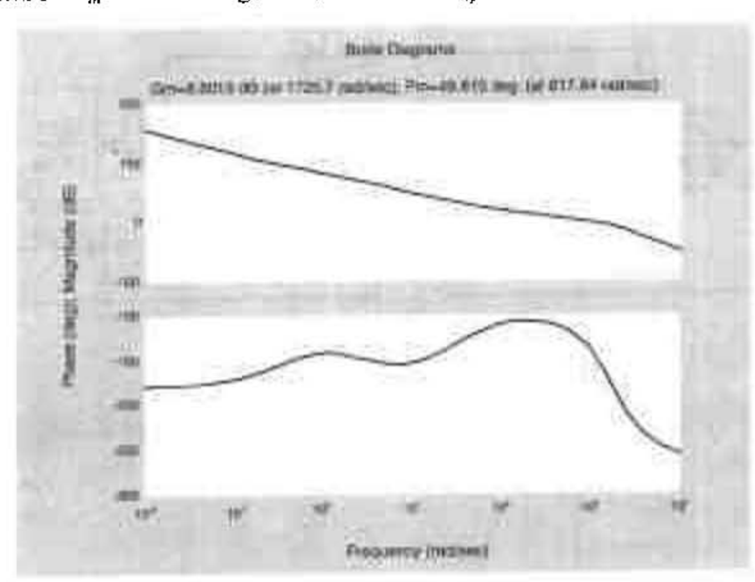


图 5-61　转速—电流环的 Bode 图

（3）对位置伺服系统进行仿真

①绘制带参数伺服系统结构图如图 5-62 所示。

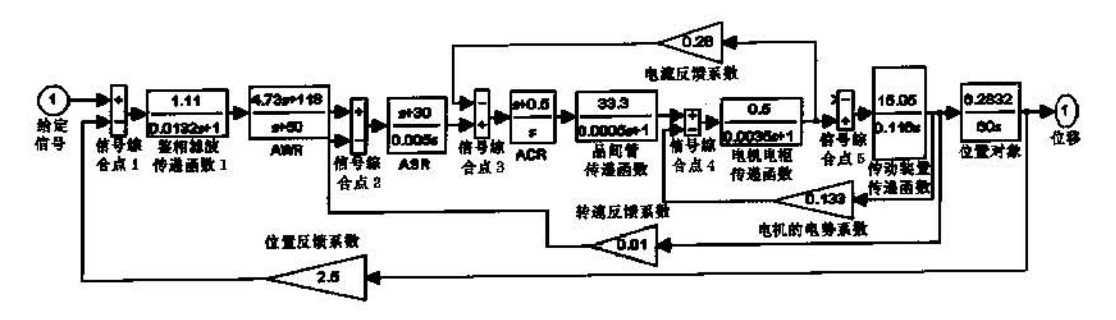


图 5-62　带参数的位置伺服三环系统结构图

②对位置伺服系统进行时域仿真。仿真程序段如下：

```
[a,b,c,d]=linmod2('t5327');s1=ss(a,b,c,d);sys=tf(s1);
t=[0:0.001:1];step(sys,t)
[y,t]=step(sys); [sgm,tp,ts]=perf(1,y,t);
```

程序段运行后得到位置伺服系统阶跃给定响应曲线（如图 5-63 所示）及其时域响应性能指标：

超调量：*Sigma* = 13.9521 %
峰值时间：*tp* = 0.0335 s
调节时间：*ts* - 0.0423 s

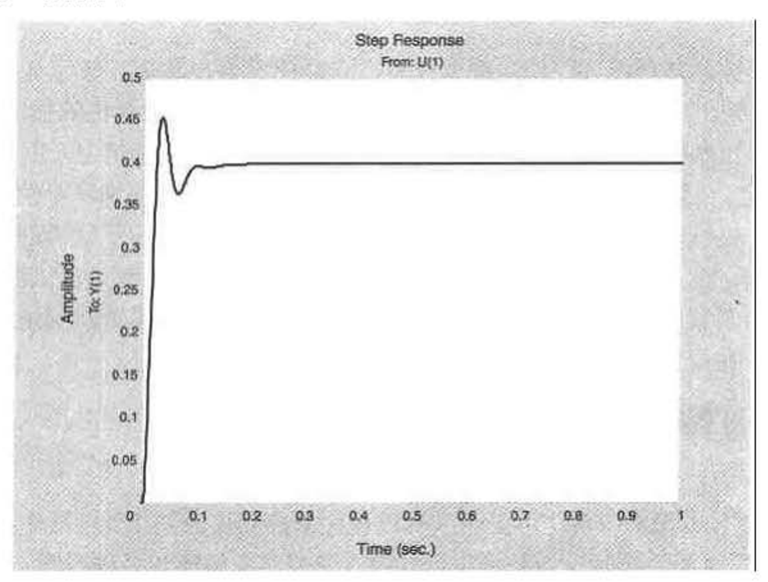


图 5-63 位置伺服系统的阶跃给定响应曲线

③ 对位置伺服系统进行频域仿真。仿真程序中用的 t5327a.mdl 模型就是位置伺服系统（图 5-62）的开环结构图。经仿真可得到伺服系统的 Bode 图（图 5-64）及其频域性能指标：

模稳定裕度 G_m=17.59 dB；　-π 穿越频率 ω_{cg} = 281.3 s^{-1}；
相稳定裕度 P_m=50.411 deg；　剪切频率 ω_{cp} = 84.954 s^{-1}

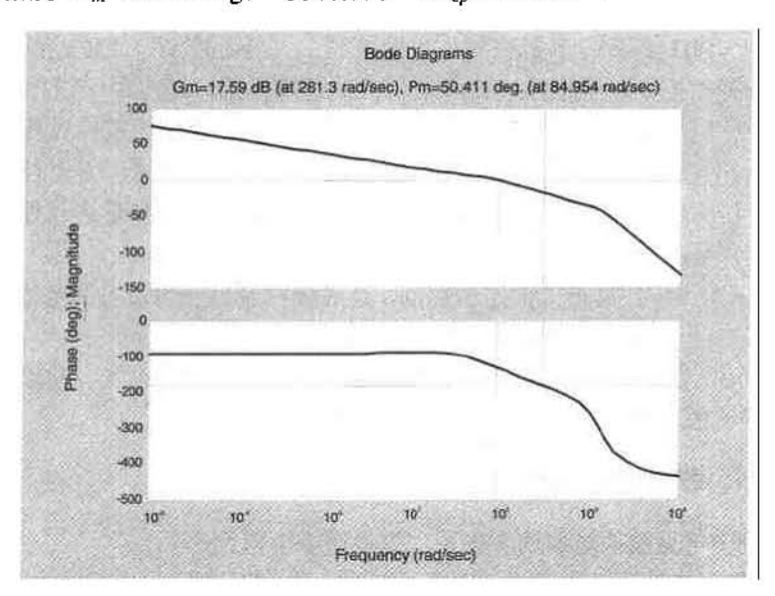


图 5-64 位置伺服系统的 Bode 图

5.4 过程控制系统的 MATLAB 计算及仿真

生产过程控制是利用过程检测控制仪表、自动化设备与装置、数字计算机等自动化技术工具，对整个生产过程进行自动检测与控制，以期达到各种最优的技术、经济指标。工业生产过程自动控制系统（简称过程控制系统），是系统的被调量为温度、压力、流量、液位（或物位）、成分、物性（氢离子浓度即 pH 值）等这样一些物理量的控制系统

在冶金、机械、石油、化工、电力、轻工、建材、原子能与环保等部门生产中，大量需要对温度、压力、流量、液位、成分等物理量实现自动控制。生产过程自动控制是一门内容极为丰富的综合性应用技术学科。它与工程实践联系紧密，与电力拖动自动控制系统一样，在现代工业生产自动化中也得到了十分广泛的应用。

5.4.1 生产过程控制的特点

与拖动自动控制系统相比，生产过程控制有以下主要特点。

（1）除受控对象外，系统由工业系列过程检测控制仪表（甚至包括计算机）所组成。

（2）受控对象的实际设备是确定已知的。受控对象的形式多种多样，一般都具有惯性大、滞后大，而且大多还有非线性特性，因此被控过程的形式也是多种多样的。例如热工过程中的锅炉、热交换器、核动力反应堆；冶金过程中的转炉、平炉；机械工业中的热处理炉；石油化工过程中的精馏塔、化学反应器、流体传输设备等。这些对象工作机理比较复杂，它们的动态特性是未知的或者是不十分清楚的，很难用解析方法得出其精确的动态数学模型。

（3）由于现代工业生产的工艺条件很复杂，对过程控制的要求很高，被控过程形式具有多样性，因而过程控制应用的方案也具有多样性。以控制的角度而论，各种控制诸如单回路控制、PID 控制、串级控制、前馈控制、大延迟控制、比值控制、均匀控制、分程控制、选择性控制、解耦控制、多变量控制等等，都广泛应用于过程控制中。

（4）控制过程是缓慢过程，且多半属于参量控制。这是由于被控过程的大惯性与大滞后的特点所决定的。为了量化说明生产过程被控制得是否正常，需要对表征生产过程特点的温度、压力、流量、液位（或物位）、成分、物性等过程参量进行控制，故而控制过程多半属于参量控制。

（5）负反馈闭环控制形式的定值控制，是过程控制里应用最多的基本控制形式。在运行时，系统对于被调量的给定值是固定不变的。

本节仅就生产过程自动控制的单回路控制系统、串级控制系统、前馈控制系统、大延迟控制系统等的 MATLAB 计算与仿真进行介绍。

5.4.2 过程控制被控对象的动态特性

过程控制中常见被控对象的动态特性大多有以下特征：

（1）被控对象的阶跃响应曲线是不振荡的、单调衰减的；被调量的变化非常缓慢，时间往往以数分钟甚至于数小时计，其过渡过程大多属于慢过程。

（2）被控对象动态特性有迟延（也叫延迟或滞后）。因为对象的大设备、大存储容量、大惯性及阻力，使被控参数不可能立即响应而有延迟。时间延迟有传输延迟与容量延迟之分。传输延迟（也称纯延迟）是因为物料或能量需要经过一个传输过程而造成的；而容量延迟则是由于对象中包含有多个容积所造成的。当过程的纯延迟时间τ与其动态时间常数T的关系满足$\tau/T\geqslant0.3$时，则被控对象就被认为是较大延迟对象了。

（3）被控对象可以是自平衡的或者是非自平衡的。被控对象受到扰动作用后系统平衡状态被破坏，无须外加任何控制作用，依靠对象本身趋向平衡的特性，能自动达到新的平衡状态者，被控对象被称为自平衡的；否则是非自平衡的。

（4）被控对象动态特性常有非线性特性。

5.4.3 过程控制系统的性能指标

过程控制系统的性能指标，除了有与拖动自动控制系统相同的部分外，还有过程控制系统特有的。

1．系统阶跃响应的性能指标

过程控制系统的性能指标，也是由系统阶跃响应曲线变化的一些特征量来定义的。这一类性能指标是与拖动自动控制系统相同或相近的。

（1） 余差（残余偏差或静态偏差）

动态过程结束进入平衡工作状态后，被调量新的稳态值$c(\infty)$与给定值$r(t)$的稳态之差就是系统偏差：$e(t)=r(t)-c(\infty)$。它是过程控制系统稳态准确性的衡量指标，一般要求余差越小越好，即最好为零或者不超过预定值。拖动控制系统中则叫静差或者稳态误差。

（2）衰减比与衰减率

系统阶跃响应曲线最先两个相邻的同向波峰值之比叫做衰减比（参见图 5-65），即

$$衰减比\qquad n=\frac{B_1}{B_2} \tag{5-1}$$

常表示为$n:1$，衰减比越大越好。

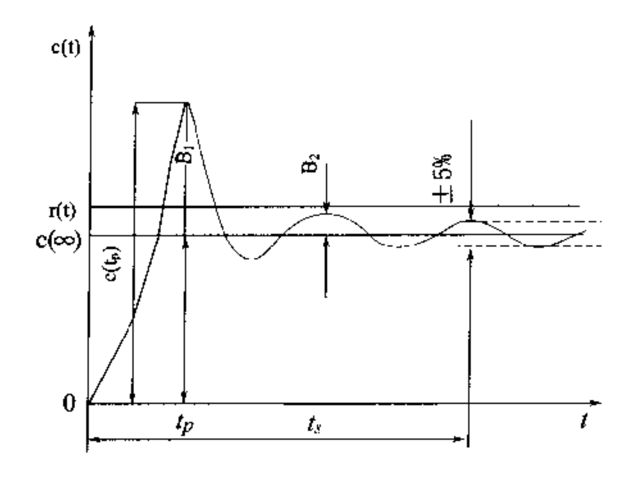


图 5-65 闭环系统的阶跃给定的响应曲线

衰减率是指系统阶跃响应曲线每经过一个周期后，波动幅度衰减的百分数，即

$$\text{衰减率}\quad \psi=\frac{B_1-B_2}{B_1} \tag{5-2}$$

衰减率越接近于 1 表示衰减得越快，则越好。

衰减比与衰减率 都是衡量振荡过程衰减程度的指标，且两者有简单的对应关系，比如衰减比为 $n:1=4:1$，即相当于衰减率 $\psi=0.75$。在过程控制中，为保证系统有一定的稳定裕量，一般要求衰减比 $n:1=4:1\sim10:1$，即衰减率 $\psi=0.75\sim0.9$。这样的系统大约经过两个周期后就看不出振荡了。

（3）最大动态偏差与超调量

系统阶跃响应过渡过程开始后第一个波峰超过其新稳态值的幅度叫做最大动态偏差，如图中的 B_1。最大动态偏差占被调量稳态值变化幅度的百分数称为超调量，即

$$\sigma\,\%=\frac{c(t_p)-c(\infty)}{c(\infty)}\cdot100\,\% \tag{5-3}$$

最大动态偏差与超调量都是越小越好，这些都与拖动自动控制系统中的概念完全一样。

（4）调节时间与振荡频率

调节时间 t_s 理论上是从过渡过程开始到结束所需要的时间；实际上是从过渡过程开始到被调量进入新稳态值的 ±5%（或 ±2%）范围内所经历的时间 。调节时间是衡量系统快速性能的一个指标，要求 t_s 值越小（即调节时间越短）越好。系统阶跃响应过渡过程大都为衰减振荡过程。既然是振荡，就有振荡频率。即振荡频率 f 为单位时间内振荡的周波数。振荡频率的倒数则为振荡周期 T。

2. 系统偏差积分性能指标

这一类性能指标是过程控制系统所特别关注的。它是过渡过程中被调量偏离其新稳态值之偏差沿时间轴的积分。偏差幅值的增大或者是时间的拖长都会使偏差积分的值增大，控制过程希望这个积分值越小越好。系统偏差积分性能指标是以目标函数形式表示的，常用的有以下几种形式：

（1）偏差积分（IE）

$$\mathrm{IE}=\int_0^\infty e(t)\mathrm{d}t \tag{5-4}$$

（2）绝对偏差积分（IAE）

$$\mathrm{IAE}=\int_0^\infty |e(t)|\mathrm{d}t \tag{5-5}$$

（3）平方偏差积分（ISE）

$$\mathrm{ISE}=\int_0^\infty e^2(t)\mathrm{d}t \tag{5-6}$$

（4）时间绝对偏差乘积积分（ITAE）

$$\mathrm{ITAE}=\int_0^\infty t|e(t)|\mathrm{d}t \tag{5-7}$$

（5）时间偏差平方乘积积分（ITSE）

$$\mathrm{ITSE}=\int_0^\infty te^2(t)\mathrm{d}t \tag{5-8}$$

以上各式中，系统偏差$e(t)=r(t)-c(\infty)$，参见图 5-65。

5.4.4 过程控制中延迟特性的处理

过程控制的对象多是大设备、大存储容量、大惯性及阻力，使系统的被调量不可能立即响应而有延迟。这种缓慢过程比起拖动控制系统要突出与明显得多。在此有必要简单回顾一下自动控制理论中的延迟特性。

1．控制系统中延迟特性的传递函数模型

自动控制原理中已经学过，延迟特性的方程为：

$$c(t)=r(t-\tau) \tag{5-9}$$

式中，$r(t)$是输入量；$c(t)$是输出量；τ 是延迟时间。

则延迟特性的传递函数为：

$$G(s)=\frac{C(s)}{R(s)}=\mathrm{e}^{-\tau s} \tag{5-10}$$

由延迟特性与典型环节的不同组合，即构成各种不同类型传递函数的典型工业过程。

（1）带纯延迟的一阶惯性环节

$$G(s)=\frac{K}{Ts+1}e^{-\tau s} \tag{5-11}$$

（2）带纯延迟的二阶或 n 阶惯性环节

$$G(s)=\frac{K}{(T_1s+1)(T_2s+1)}e^{-\tau s} \tag{5-12}$$

或

$$G(s)=\frac{K}{(Ts+1)^n}e^{-\tau s} \tag{5-13}$$

（3） 带纯延迟的有理分式表示的传递函数

$$G(s)=\frac{b_ms^m+\cdots+b_1s+b_0}{a_ns^n+\cdots a_1s+a_0}e^{-\tau s} \qquad (n\geqslant m) \tag{5-14}$$

（4）非自平衡过程的传递函数里应有一个积分环节，例如有：

$$G(s)=\frac{K}{T_as}e^{-\tau s} \tag{5-15}$$

和

$$G(s)=\frac{K}{T_as(Ts+1)}e^{-\tau s} \tag{5-16}$$

已经知道，纯延迟时间函数可以用控制系统工具箱中的 pade（ ）函数命令来近似求取其传递函数。对于$e^{-\tau s}$，其 pade ( ) 函数的调用格式为：

$$[np,dp]=\mathbf{pade}(tau,n)$$

式中 *tau* 即是延迟常数τ，*n* 为 pade ( ) 函数近似的阶次，近似的传递函数分子、分母在[*np*,*dp*]变量中返回，即有：

$$e^{-\tau s} \approx P_{n,\tau}(s)=tf(np,dp) \tag{5-17}$$

2．带延迟特性闭环系统的近似模型

带有时间延迟环节的反馈控制系统，其典型结构如图 5-66 所示。

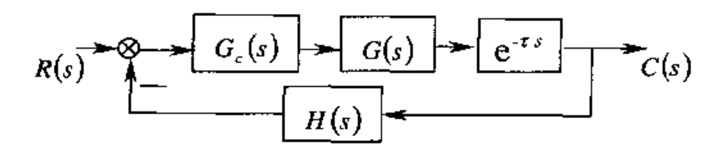


图 5-66　带有延迟环节的反馈系统典型结构图

在带有延迟环节的系统中，将延迟环节用相应的 pade（）函数近似，图 5-66 所示的系统的闭环传递函数模型为：

$$G_{cl}(s)=\frac{G_c(s)G(s)e^{-\tau s}}{1+G_c(s)G(s)H(s)e^{-\tau s}} \approx \frac{G_c(s)G(s)P_{n,\tau}(s)}{1+G_c(s)G(s)H(s)P_{n,\tau}(s)} \tag{5-18}$$

在实际应用中，这种模型经常会在 $0\sim\tau$ 时间间隔内出现微弱振荡，对于系统仿真计算的准确性是极为不利的。 此弊端解决的办法如下。

若只对分母中的$e^{-\tau s}$作 pade（）函数近似，则上述闭环传递函数模型就近似为：

$$G_{cl}(s)=\frac{G_c(s)G(s)e^{-\tau s}}{1+G_c(s)G(s)H(s)e^{-\tau s}} \approx \frac{G_c(s)G(s)e^{-\tau s}}{1+G_c(s)G(s)H(s)P_{n,\tau}(s)} \tag{5-19}$$

这个传递函数模型可以用下图 5-67 来表示。

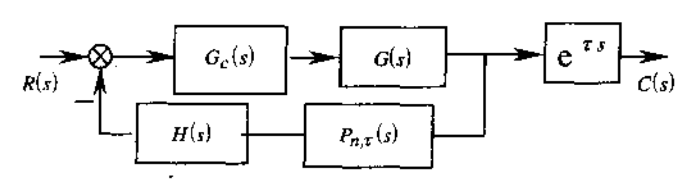


图 5-67　带有延迟环节反馈系统的近似结构图

图 5-67 所示的近似结构图的闭环部分，可按照闭环连接等效变换法则，求其等效模型。其后串联的$e^{-\tau s}$部分，可以用控制系统工具箱中的另一个 set ( ) 函数来实现。在 4.1 节中已经介绍过 set ( ) 函数，set ( ) 函数的功能是设置或者修改线性时不变（LTI）对象的属性值。 set ( ) 函数的调用格式为：

set (sys,'Property',Value)

set (*sys*,*'Property'*,*Value*)设置 LTI 对象 *sys* 的 *Property* 属性的属性值为 *Value*。*Property*

为任意 LTI 对象支持的属性名字符串，可以为属性的全名，也可以为大小写不敏感无歧义的字符串缩写。

LTI 对象属性中有个输入延时属性 *Td*，如果把图 5-67 的闭环部分当作 LTI 对象，将其 *Td* 设置为τ，即可实现带有延迟环节反馈系统之等效模型的仿真。此时的 set ( ) 函数为：

set (Gc,'Td',tau)

Gc 为图5-67 所示结构图闭环部分的等效传递函数，*Td* 为对象Gc 中的输入延时属性，*tau* 即是延迟常数τ。

5.4.5 简单回路控制系统的 MATLAB 计算及仿真

过程控制中的单回路控制即简单控制系统，是由广义对象与调节器构成的负反馈闭环控制系统。单回路控制系统由于结构简单、投资少、易于调整，又能满足一般生产过程的工艺要求，应用非常广泛。直到现在为止，工程实际中单回路控制系统仍占过程控制总量的 85% 以上。

在此举两例说明对过程控制中简单控制系统进行 MATLAB 计算仿真的方法与过程。

【例 5-13】已知过程控制系统的被控广义对象为一个带延迟的惯性环节，其传递函数为：

$$G_0(s)=\frac{1}{20s+1}e^{-2.5s}$$

设定控制所用串联校正 PI 调节器传递函数为：

$$G_c(s)=\frac{48.74\,s+6.062}{8.04\,s}$$

试对单位负反馈系统进行阶跃给定响应的仿真，并计算该系统跟随过程的性能指标。

【解】

为进行系统阶跃响应仿真并计算其性能指标，给出如下 MATLAB 程序 L5401.m。程序里调用了作者开发的函数 perf( )、ster( )与 targ( )。

```
% MATLAB PROGRAM L5401.m
%
n1=[1];d1=[20 1];G1=tf(n1,d1);
tau=2.5;[np,dp]=pade(tau,2);Gp=tf(np,dp);
n2=[48.74 6.062];d2=[8.04 0];G2=tf(n2,d2);
sys=feedback(G1*G2,Gp);
set(Gc,'Td',tau);step(sys)
[y,t]=step(sys);[sigma,tp,ts]=perf(1,y,t);
[pm,tp,ess,sigma,n,pusi,T,f]=targ(y,t);
t=[0:0.1:300]';[ess1]=ster(C,sys,t)
[ess2=ster(1,sys,t)
```

序运行后有图 5-68 所示的阶跃给定响应仿真曲线，并计算出以下性能指标：

过程控制的超调量：*sigma* = 52.9794 %

过程控制的峰值时间：*tp* = 8.9409 s

过程控制的调节时间（5%）：ts = 19.2574 s

过程控制的余差：ess = 2.2204e 016　　(300s 实际余差为 0)

过程控制的单位速度响应余差：ess = 1.3263

过程控制的第一（正向）波峰值：$b1$ = 0.5317

过程控制的第二（正向）波峰值：$b2$ = 0.0404

过程控制的衰减比：n = 13.1489

过程控制的衰减率：$pusi$ = 0.9239

过程控制的衰减振荡周期：T = 15.4747 s

过程控制的振荡频率：f = 0.0646 s^{-1}

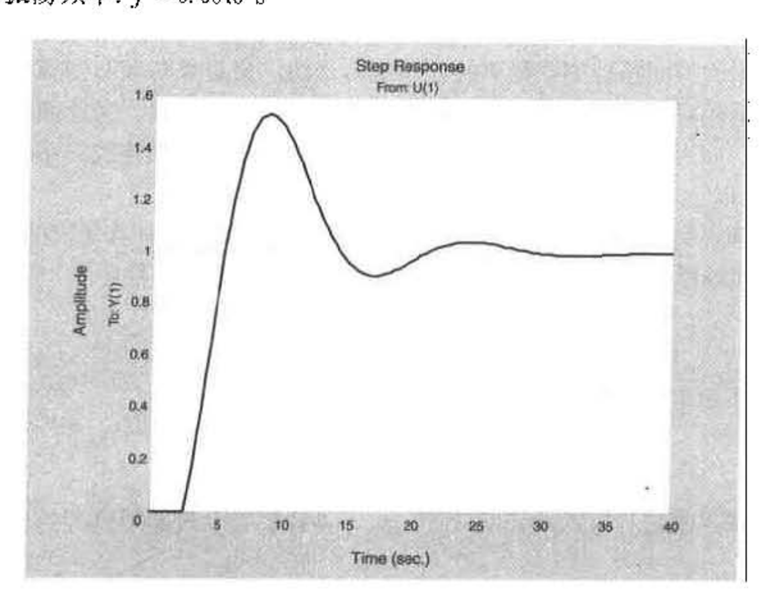


图 5-68　单回路系统阶跃给定响应仿真曲线

【例 5-14】已知过程控制系统的被控广义对象传递函数为：

$$G_0(s)=\frac{1}{(5s+1)(2s+1)(10s+1)}$$

设定控制所用串通校正 PID 调节器传递函数为：

$$G_c(s)=\frac{100s^2+40s+5}{1.6s^2+10s}$$

试对单位负反馈系统进行阶跃给定响应的仿真，并计算该系统阶跃给定响应的性能指标。

【解】

为进行系统阶跃响应的仿真并计算其性能指标，给出 MATLAB 程序 L5402.m 如下。

```
% MATLAB PROGRAM L5402.m
%
global y t
n1=[1];;d1=conv([5 1],conv([2 1],[10 1]));G1=tf(n1,d1);
```

```
n2=[100 40 5];d2=[1.6 10 0];G2=tf(n2,d2);
sys=feedback(G1*G2,1);step(sys)
[y,t]=step(sys); [sigma,tp,ts]=perf(1,y,t);
[pm,tp,ess,sigma,n,pusi,T,f]=targ(y,t);
t=[0:0.1:300]';[ess1]=ster(0,sys,t)
[ess2]=ster(1,sys,t)
```

程序运行后有图 5-69 所示的阶跃给定响应仿真曲线，并计算出以下性能指标。

过程控制的超调量：*sigma* = 32.0754 %

过程控制的峰值时间：*tp* = 13.0626 s

过程控制的调节时间（5%）：*ts* = 21.0137 s

过程控制的余差：*ess* = 3.9524e-014（300s 实际余差为 0）

过程控制的单位速度响应余差：*ess* = 2

过程控制的第一（正向）波峰值：*b1* = 0.3196

过程控制的第二（正向）波峰值：*b2* = 0.0204

过程控制的衰减比：*n* = 15.6772

过程控制的衰减率：*pusi* = 0.9362

过程控制的衰减振荡周期：*T* = 30.6687 s

过程控制的振荡频率：*f* = 0.0326 s^{-1}

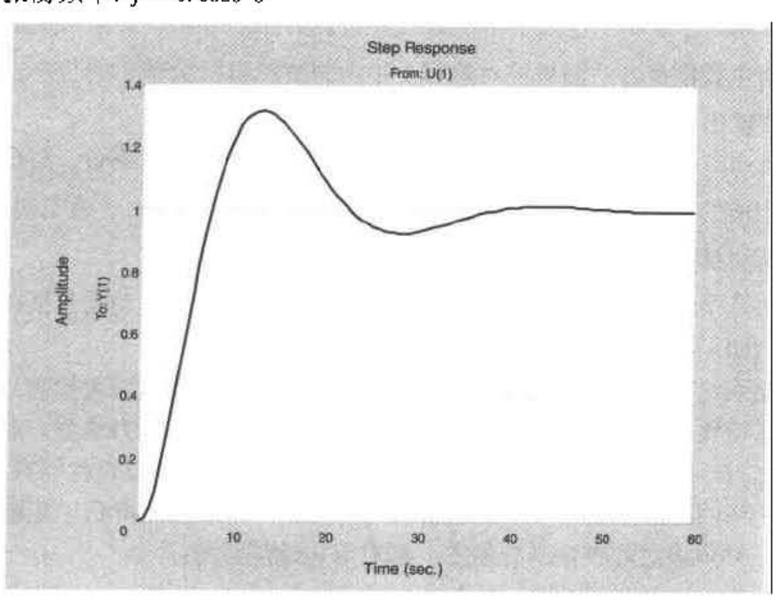


图 5-69　简单系统阶给定跃响应仿真曲线

5.4.6　串级控制系统的 MATLAB 计算及仿真

在单回路控制系统满足不了生产工艺高要求的情况下，串级过程控制系统便应运而生。串级过程控制系统与电力拖动自动控制系统中的电流、转速双闭环调速系统相同，其系统特点与分析方法也基本相似。

在过程控制中，有些专用技术术语与电力拖动自动控制系统不大一样，需要弄清楚其基本概念。串级控制系统的一般结构如图 5-70 所示。

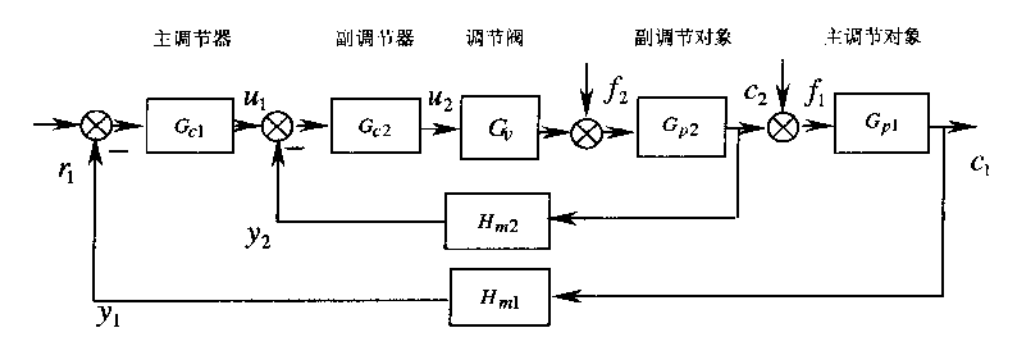


图 5-70　串级控制系统的框图

说明以下基本概念时，借用了 图 5-70 中设定的变量、文字标识。

主被控参量即系统的输出量 c_1；副被控参量即系统副回路的输出量 c_2；主调节器 G_{c1} 是按主被控量的测量值 y_1 与给定值 r_1 的偏差作输入的调节器，其输出量 u_1 作为副调节器的给定值；副调节器 G_{c2} 是按主调节器输出 u_1 与副被控量 c_2 的测量值 y_2 的偏差作输入的调节器，其输出量 u_2 直接控制调节阀 G_v 动作。

主被控对象 G_{p1} 是由主被控参量 c_1 表征其特性的生产对象，其输入量是副被控参量 c_2，输出量为主被控量 c_1；副被控对象 G_{p2} 是由副被控量 c_2 为输出的生产过程，其输入量是调节阀的输出。

主回路是由主副调节器 G_{c1} 与 G_{c2}、调节阀 G_v、主副被控对象 G_{p1} 与 G_{p2}、主测量变送器 H_{m1} 组成的闭环回路，即外环回路；副回路是副调节器 G_{c2}、调节阀 G_v、副被控对象 G_{p2}、副测量变送器 H_{m2} 组成的闭环回路，即内环回路。

一次扰动是不包括在副回路内的所有扰动，例如 f_1；二次扰动是包括在副回路内的所有扰动，例如 f_2。

串级控制系统由于多了一个副回路，使得系统的性能得到了很大提高。它改善了被控过程的动态特性，提高了系统的工作频率，使得系统抗扰动能力增强，对于负载或操作条件的变化有一定的自适应能力。由于串级控制比简单闭环负反馈控制有显著的优点，因此在简单闭环回路控制满足不了工艺要求的情况下，串级控制得到了普遍应用。

【例 5-15】已知串级过程控制系统的主、副对象传递函数为：

$$G_{p1}(s)=\frac{1}{(30s+1)(3s+1)}\text{；}\quad G_{p2}(s)=\frac{1}{(10s+1)(s+1)^2}$$

设定控制所用主、副调节器传递函数分别为：

$$G_{c1}(s)=Kc1\left(1+\frac{1}{T_i s}\right)=8.4\left(1+\frac{1}{12.8s}\right)$$

$$G_{c2}(s)=Kc2=10$$

其系统的结构如图 5-71 所示：

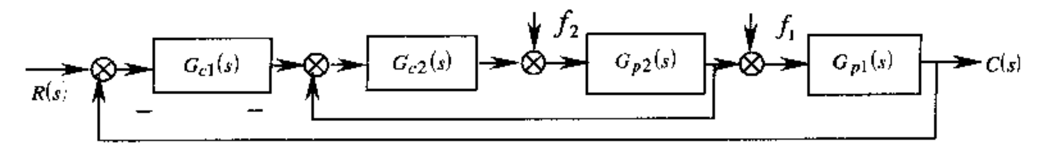


图 5-71　串级过程控制系统的结构框图

试对系统进行仿真，并计算该系统过程控制的性能指标。

【解】

（1）将已知参数带入并绘制串级过程控制系统的 SIMULINK 结构图如图 5-72 所示。

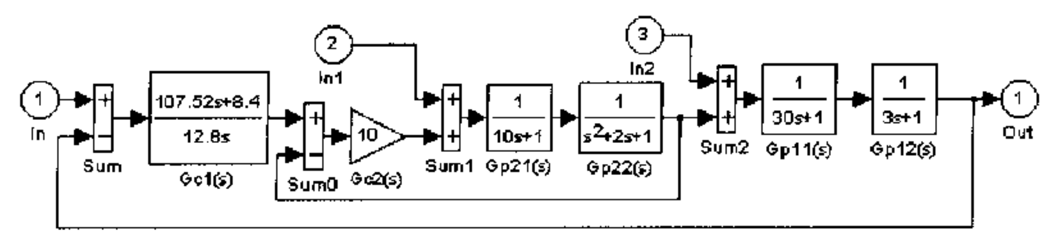


图 5-72　带实际参数的串级系统 SIMULINK 结构图

（2）串级系统阶跃给定响应过程仿真

为实现串级系统阶跃响应过程仿真，给出如下 MATLAB 程序 L5403a.m。程序中 t5408.mdl 即图 5-72，但无扰动信号作用②与③。

```
% MATLAB PROGRAM L5403a.m
%
[a,b,c,d]=linmod2('t5408');sys1=ss(a,b,c,d); step(sys1);
[y,t]=step(sys1);perf(1,y,t);
[mp,tp,b1,b2,sigma,n,pusi,T,f]=targ(y,t);
sys=tf(sys1);t=[0:0.01:800]';
[ess1]=ster(0,sys,t);[ess2]=ster(1,sys,t);
```

程序运行后有图 5-73 所示的阶跃给定响应仿真曲线，并计算出以下性能指标：

过程控制的超调量：*sigma* = 51.7811 %

过程控制的峰值时间：*tp* = 12.3233 s

过程控制的调节时间（5%）：*ts* = 28.7544 s

过程控制的余差：*ess* = -1.5321e-014　　（此为截断误差引起 800s 余差实际为 0）

过程控制的单位速度响应余差：*ess* = 1.6762

过程控制的第一（正向）波峰值：*b1* = 0.5211

过程控制的第二（正向）波峰值：*b2* = 0.0516

过程控制的衰减比：*n* = 10.0969

过程控制的衰减率：*pusi* = 0.9010

过程控制的衰减振荡周期：*T* = 23.4730 s

过程控制的振荡频率：*f* = 0.0426 s^{-1}

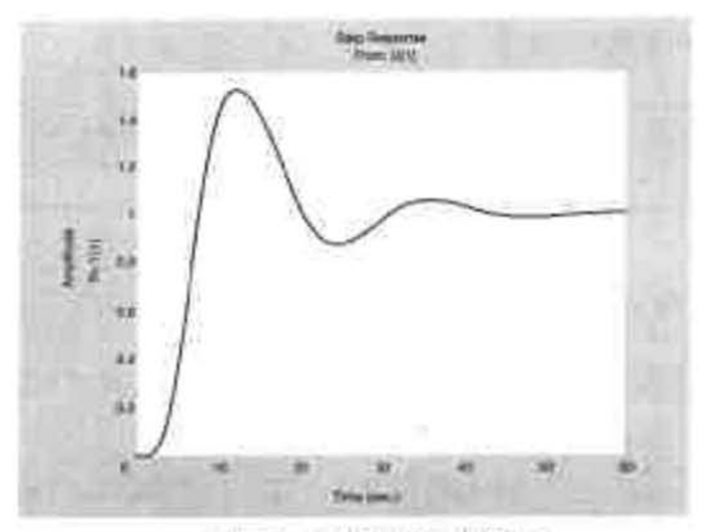


图 5-73 串级系统阶跃响应曲线

（3）扰动作用②的阶跃扰动过程仿真

系统在扰动作用②的阶跃扰动过程仿真由以下 MATLAB 程序 L5403b.m 完成。程序里调用了作者开发的 MATLAB 函数 dist( )。程序中，t5408a.mdl 就是图 5-72，但无给定信号作用①与扰动信号作用③。

```
% MATLAB PROGRAM L5403b.m
%
[a,b,c,d]=linmod2('t5408a');sys1=ss(a,b,c,d); step(sys1)
[y,t]=step(sys1);[detac,tp,tv]=dist(1,y,t);
[detac,tp,tv]=dist(2,y,t);
```

程序运行后有图 5-74 所示的阶跃扰动响应仿真曲线，并计算出以下性能指标。

最大动态降落：*detac* = -0.0130

最大动态降落时间：*tp* = 10.1745 s

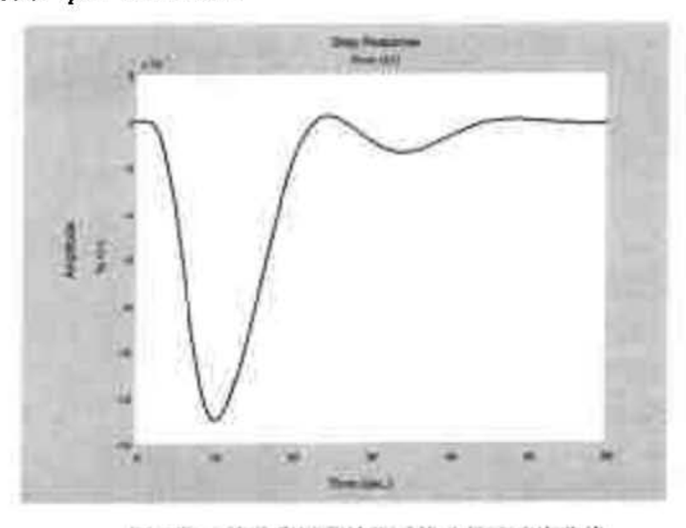


图 5-74 扰动作用②的阶跃扰动扰动响应曲线

从图 5-74 中看出，对于阶跃扰动响应仿真曲线纵坐标，最大降落为 $13\times10^{-4}=0.0013$，程序 dist.m 的误差带 5%或 2%已远远大于此最大降落，检测不出 tv 了。如将程序 dist( )的误差带设定为 5‰或 2‰，程序运行结果为：

基准值 5‰范围的恢复时间：tv = 17.3400 s

基准值 2‰范围的恢复时间：tv = 19.7900 s

（4） 扰动作用③的阶跃扰动过程仿真

系统在扰动作用点③的阶跃扰动过程仿真由以下 MATLAB 程序 L5603c.m 完成。程序中，t5408b.mdl 就是图 5-72，但无给定信号作用①与扰动信号作用②。

```
% MATLAB PROGRAM L5603c.m
%
[a,b,c,d]=linmod2('t5408b');sys1=ss(a,b,c,d);
t=[0:0.01:80];
step(sys1);
[y,t]=step(sys1,t);
[detac,tp,tv]=dist(1,y,t);
[detac,tp,tv]=dist(2,y,t);
```

程序运行后有图 5-75 所示的阶跃扰动响应仿真曲线，并计算出以下性能指标：

最大动态降落：$detac$ = -0.1319

最大动态降落时间：tp = 9.5100 s

基准值 5%范围的恢复时间：tv = 16.6600 s

基准值 2%范围的恢复时间：tv = 19.2000 s

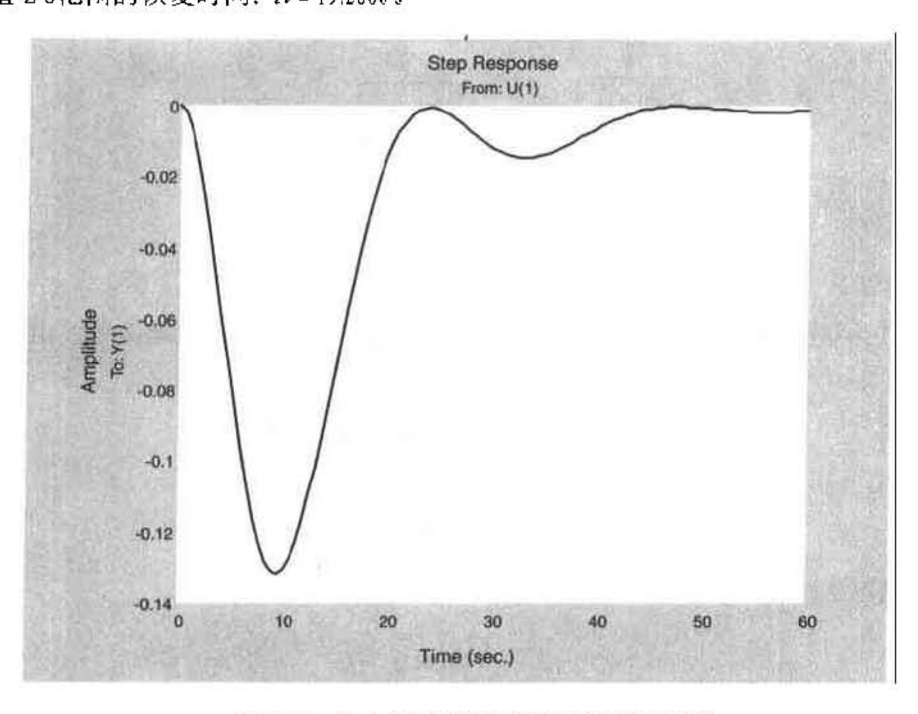


图 5-75　扰动作用③的阶跃扰动扰动响应曲线

5.4.7 前馈控制系统的 MATLAB 计算及仿真

1. 前馈控制简述

负反馈闭环控制的特点是当被控制过程受到扰动作用后，一直要到被调量出现偏差时，经检测传输到给定输入端并进行比较、运算后，再通过调节器动作，才能补偿扰动对被调量的影响。

前馈控制的思路是这样的。系统被调量除了跟随给定量变化而应该变化外，被调量还会产生偏差，其原因是由于扰动作用引起的。倘若在扰动出现时就能立即进行控制，而不是等到偏差发生后再进行控制，这样就可以有效地消除扰动对系统被调量的影响。

前馈控制是将扰动信号经前馈控制器处理后用以消除扰动对被调量的影响，它是按扰动进行的补偿控制，所以前馈控制又叫作“扰动补偿”。由自动控制原理知道，扰动补偿属于开环控制。前馈控制对系统的稳定性无影响，只要原系统是稳定的，施以前馈控制后系统仍然稳定。

前馈控制只能对于可以测量的扰动作用进行扰动补偿。前馈控制器的结构、参数取决于被控制对象与扰动通道的特性。

2. 不变性原理

不变性原理是实现前馈控制的理论基础。“不变性”是指控制系统的被调量不受扰动作用的影响，与扰动完全无关，或在一定准确度下无关。

不变性原理是通过前馈控制器的校正作用来消除扰动对系统被调量的影响的。前馈控制的框图如图 5-76 所示:

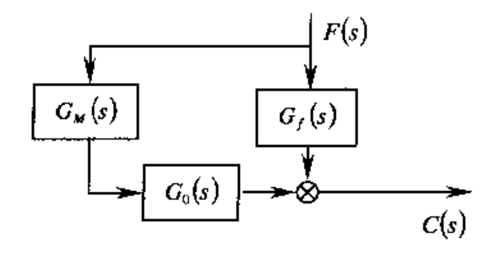


图 5-76 前馈控制系统框图

图中 $G_f(s)$ 是扰动通道的传递函数，$G_M(s)$ 是前馈校正器的传递函数，由图 5-76 可知

$$C(s)=G_f(s)F(s)+G_M(s)G_0(s)F(s) \tag{5-20}$$

则有

$$\frac{C(s)}{F(s)}=G_f(s)+G_M(s)G_0(s) \tag{5-21}$$

根据不变性原理，应有

$$\frac{C(s)}{F(s)}=0 \tag{5-22}$$

可得前馈校正器的模型为：

$$G_M(s)=-\frac{G_f(s)}{G_0(s)} \tag{5-23}$$

3．前馈控制举例

【例 5-16】已知冷凝器温度前馈－反馈复合控制系统的控制通道特性、扰动通道特性传递函数分别为：

$$G_o(s)=\frac{0.94}{55s+1}e^{-8s}；\quad G_f(s)=\frac{1.05}{41s+1}e^{-6s}$$

其系统的结构框图如图 5-77 所示：

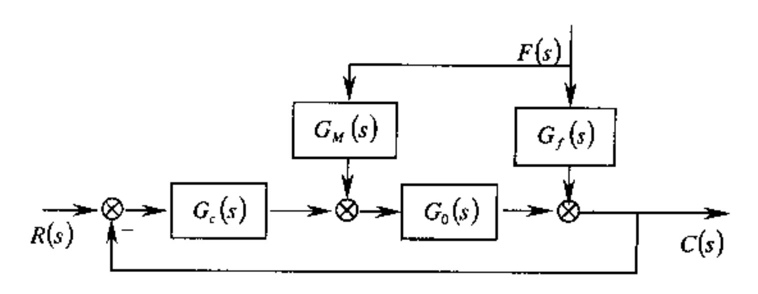


图 5-77　前馈－反馈复合控制系统结构框图

设定控制所用 PI 调节器传递函数为：

$$G_c(s)=Kc1\left(1+\frac{1}{T_i s}\right)=5.588\left(1+\frac{1}{25.54s}\right)=\frac{142.7s+5.588}{25.54\,s}$$

试对系统进行阶跃给定响应与阶跃扰动响应的仿真，并计算该系统过程控制的性能指标。

【解】

（1）处理延迟环节

为便于计算，将$G_o(s)$与$G_f(s)$中的延迟环节用 **pade**（）函数进行近似处理，其指令如下：

```
[np0,dp0]=pade(8,2);
Go2=tf(np0,dp0)
[npf,dpf]=pade(6,2);
Gf2=tf(npf,dpf)
```

指令执行后有：

```
Transfer function:
s^2 - 0.75 s + 0.1875
---------------------
s^2 + 0.75 s + 0.1875
Transfer function:
s^2 - s + 0.3333
```

```
----------------
s^2 + s + 0.3333
```

$$\text{即有}\qquad G_o(s)\approx\frac{0.94}{55s+1}\cdot\frac{s^2-0.75s+0.1875}{s^2+0.75s+0.1875}$$

$$G_f(s)\approx\frac{1.05}{41s+1}\cdot\frac{s^2-s+0.3333}{s^2+s+0.3333}$$

（2）进行阶跃给定响应的仿真

系统阶跃给定响应仿真的结构图如图 5-78 所示：

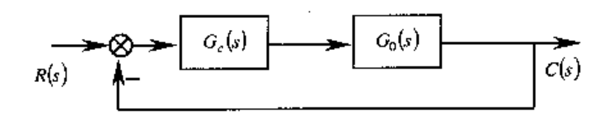


图 5-78 反馈控制系统结构框图

为进行系统阶跃响应仿真，并计算其性能指标，给出如下 MATLAB 程序 L5404a.m。

```
% MATLAB PROGRAM L5404a.m
%
n1=[0.94];d1=[55 1];G1=tf(n1,d1);
tau=8;[np,dp]=pade(tau,2);Gp=tf(np,dp);
n2=[142.7 5.588];d2=[25.54 0];G2=tf(n2,d2);
Gc=feedback(G1*G2,Gp);set(Gc,'Td',tau);step(Gc)
[y,t]-step(Gc);[sigma,tp,ts]=perf(1,y,t);
[mp,tp,b1,b2,n,pusi,T,f]=targ(y,t);
sys=feedback(G1*G2*Gp,1);t=[0:0.01:200]';
[ess]=ster(0,sys,t);[ess]=ster(1,sys,t);
```

程序执行后,可得如图 5-79 所示的阶跃给定响应曲线，并计算出性能指标如下：

过程控制的超调量：*sigma* = 50.5692 %

过程控制的峰值时间：*tp* = 28.1107 s

过程控制的调节时间 (5%)：*ts* = 43.2472 s

过程控制的第一（正向）波峰值：*b*1 = 0.5070

过程控制的第二（正向）波峰值：*b*2 = 0407

过程控制的衰减比：*n* = 12.4638

过程控制的衰减率：*pusi* = 0.9198

过程控制的衰减振荡周期：*T* = 48.6531 s

过程控制的振荡频率：*f* = 0.0206 s^{-1}

单位阶跃给定响应的稳态误差：*ess*1 = 1.7466e-012　　（500s 实际约为 0 ）

单位斜坡给定响应的稳态误差：*ess*2 = 4.8624

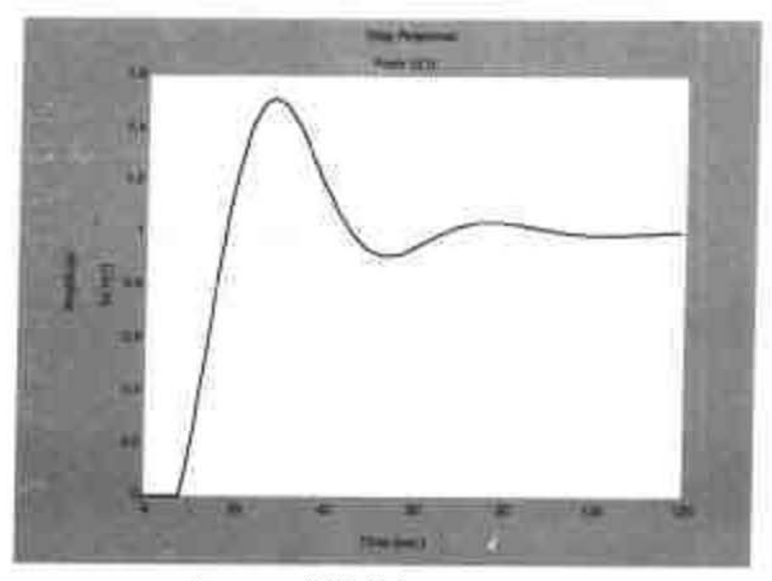


图 5-79　反馈控制系统阶跃给定响应曲线

4．进行阶跃扰动响应的仿真

对于图 5-76，根据不变性原理，要实现对扰动作用的完全补偿，须满足式（5-23）：

为方便进行阶跃扰动响应的仿真，特绘制出系统的对于扰动的 SIMULINK 结构图 5-80，并用该图给出 MATLAB 程序 L5404b.m。程序中的 t5416.mdl 即图 5-80。

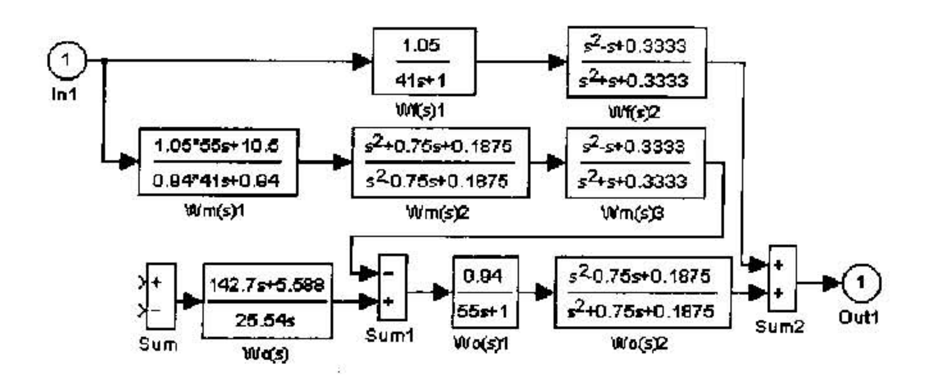


图 5-80　扰动信号作用系统的 SIMULINK 结构图

```
% MATLAB PROGRAM L5404b.m
%
[a,b,c,d]=linmod2('t5416');
sys=ss(a,b,c,d);
step(sys)
```

程序执行后，可得如图 5-81 所示的阶跃扰动响应曲线,曲线就在横坐标上，其响应值为零。这是因为根据不变性原理，实现了对扰动作用的完全补偿，即系统的被调量与对系统的扰动作用完全无关。

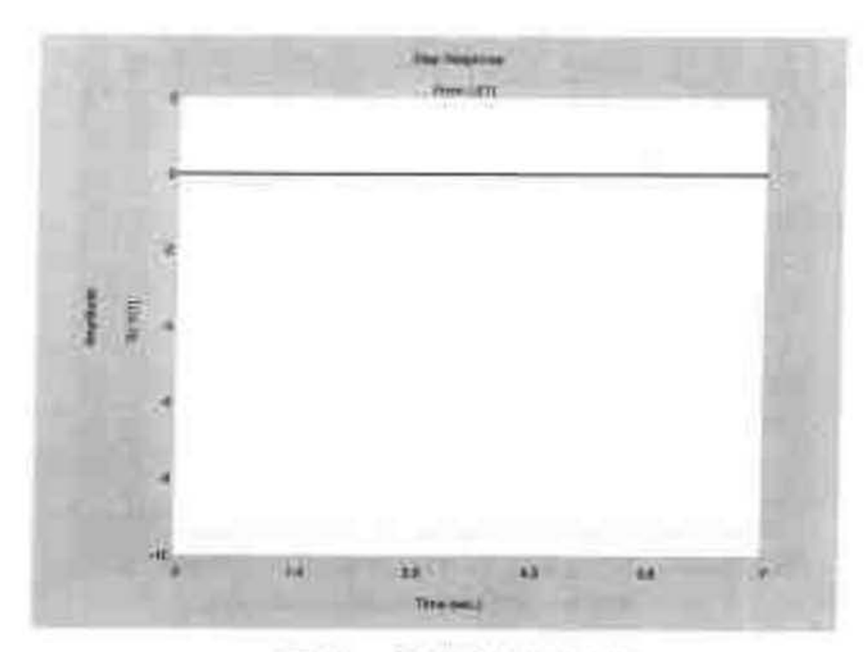


图 5-81 系统阶跃扰动响应曲线

5.4.8 大延迟系统的 MATLAB 计算及仿真

已经说明过，当过程的纯延迟时间τ与其动态时间常数 T 的关系满足$\tau/T \geqslant 0.3$ 时，则为大延迟控制系统。带有延迟特性的过程是较难控制的。当τ/T增大，控制过程中的相位滞后也增大。使得被调量不能及时反映系统所遇到或承受的扰动，即使检测信号到达调节器使之动作，也需要经延迟时间τ后，才会改变被调量使系统得到控制。于是，系统控制过程必然会经过较长的调节时间并产生明显的超调。带延迟特性过程控制的难度随着延迟时间τ的增大而加大。

大延迟控制系统一直都是控制界特别关注的课题。虽然国内外不少学者、科技工作者做了大量的研究，但是成功地应用于工程实践中的具体实例却没有多少。大延迟控制系统的控制方法，在许多过程控制的专著里都有详尽的论述。这里仅对大延迟控制系统的一般 PID 控制与史密斯（Smith）预估器控制进行 MATLAB 计算及仿真。

史密斯预估器控制的基本思路是，预先估计出过程在基本扰动下的动态特性，然后由预估器进行补偿控制，力图使被延迟了τ的被调量提前反映到调节器，并使之动作，以此来减小超调量与加速调节过程。其控制系统原理图如图 5-82 所示。

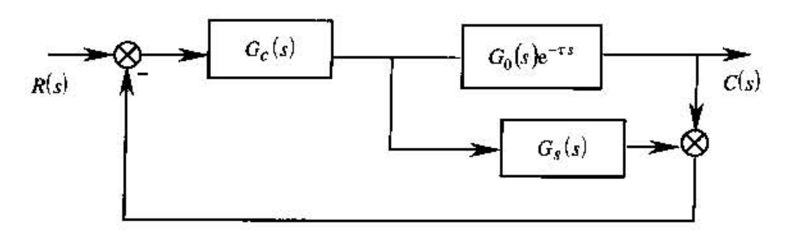


图 5-82 史密斯预估器控制原理图

图中， $G_0(s)e^{-\tau s}$——广义对象的数学模型，其中$G_0(s)$为不包含延迟时间τ 的对象模型；

$G_c(s)$——系统的一般 PID 调节器；

$G_s(s)$——史密斯预估补偿器。

若史密斯预估补偿器的传递函数为：

$$G_s(s)=G_0(s)(1-e^{-\tau s}) \tag{5-24}$$

则图 5-82 中系统闭环传递函数为：

$$\Phi(s)=\frac{C(s)}{R(s)}=\frac{G_c(s)G_0(s)e^{-\tau s}}{1+G_c(s)G_0(s)} \tag{5-25}$$

关系式（5-25）对应的动态结构图如图 5-83 所示：

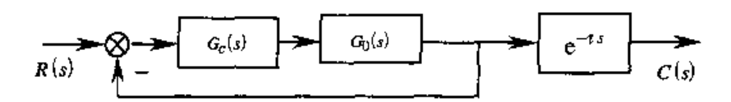


图 5-83　史密斯预估器控制原理图

与图 5-83 所示原理图等效的控制系统，很容易用 MATLAB 函数命令 set ( ) 来实现。

【例 5-17】已知大延迟系统的被控广义对象为一个带延迟的惯性环节，其传递函数为：

$$G_0(s)=\frac{2}{4s+1}e^{-4s}$$

设定控制所用 PID 调节器传递函数为：

$$G_c(s)=\frac{7.023\,s^2+4.295s+.06875}{0.9287\,s^2+6.095\,s}$$

试对系统的 PID 控制与史密斯预估器控制分别进行仿真。

【解】

（1）PID 控制阶跃给定响应仿真

为进行系统 PID 控制的阶跃给定响应仿真，给出如下 MATLAB 程序 L5405a.m。

```
% MATLAB PROGRAM L5405a.m
%
n1=[2];d1=[4 1];G1=tf(n1,d1);
tau=4;[np,dp]=pade(tau,2);Gp=tf(np,dp);
n2=[7.023 4.295 0.6875];
d2=[0.9287 6.095 0];G2=tf(n2,d2);
Gc=feedback(G1*G2,Gp);
set(Gc,'Td',tau);
step(Gc)
[y,t]=step(Gc);
[sigma,tp,ts]=perf(1,y,t);
```

程序执行后,可得如图 5-84 所示的系统 PID 控制阶跃给定响应曲线，并计算出单位阶跃给定响应的性能指标如下：

超调量：*sigma* = 26.8119 %

峰值时间：*tp* = 7.4311 s

调节时间：*ts* = 14.5083 s

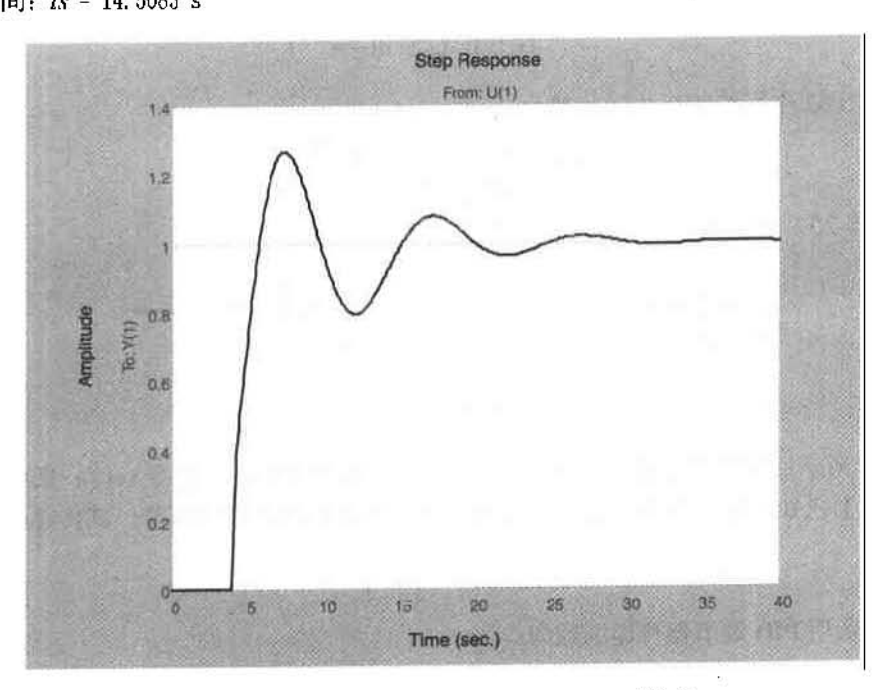


图 5-84　系统 PID 控制的阶跃给定响应曲线

（2）史密斯预估器控制的仿真

史密斯预估补偿器的传递函数必须满足式（5-24），那么史密斯预估器控制的结构框图则为图 5-83。

根据史密斯预估器控制原理，给出如下实现其仿真的 MATLAB 程序 L5405b.m。

```
% MATLAB PROGRAM L5405b.m
%
n1=[2];d1=[4 1];G1=tf(n1,d1);
tau=4;[np,dp]=pade(tau,2);Gp=tf(np,dp);
n2=[7.023 4.295 0.6875];d2=[0.9287 6.095 0];
G2=tf(n2,d2);Gc=feedback(G1*G2,1);
set(Gc,'Td',tau);step(Gc)
```

程序执行后,可得如图 5-85 所示的史密斯预估器控制的系统阶跃给定响应曲线。曲线显示史密斯预估器控制的作用完全实现，即图 5-84 所示系统 PID 控制的阶跃给定响应振荡的曲线，由预估器进行补偿控制后，使被延迟了的被调量提前反映到调节器，并使之动作，以此来减小超调使之成为单调上升的过程，预估补偿器的控制作用十分突出与明显。

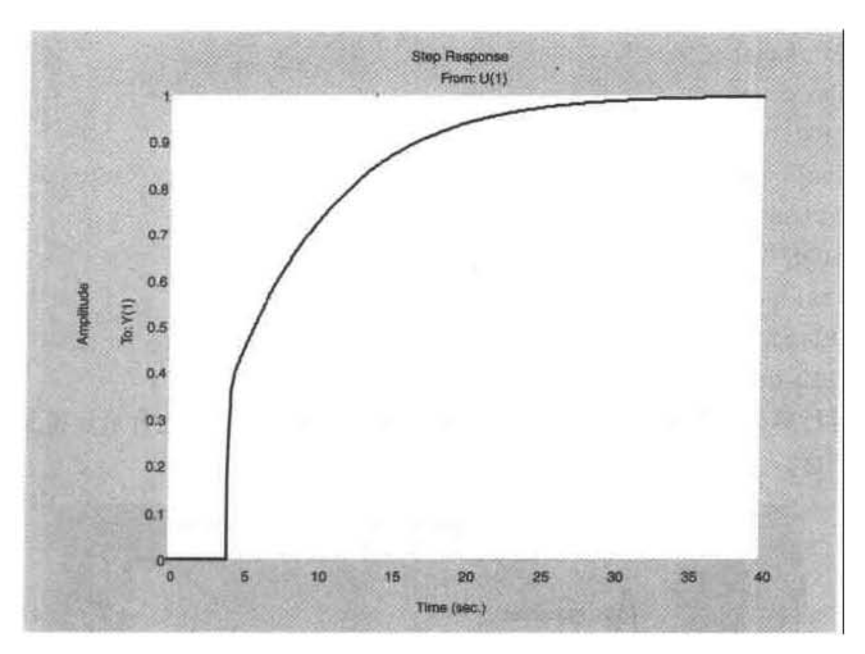


图 5-85 史密斯预估器控制的阶跃响应曲线

【例 5-18】已知钢铁厂轧钢车间加热炉传递函数模型与温度传感器及其变送器的传递函数模型分别为：

$$G_{01}(s)=\frac{9.9}{120s+1}e^{-80s};\quad G_{02}(s)=\frac{0.107}{10s+1}$$

设定控制所用 PI 与 PID 调节器传递函数分别为：

$$G_{c1}(s)=\frac{301.5s+1.228}{245.5s};\quad G_{c2}(s)=\frac{9286s^2+239.7s+1.62}{521.2s^2+144.4s}$$

试对系统的 PI 与 PID 控制与史密斯预估器控制分别进行仿真。

【解】

（1）由题意知广义被控对象的数学模型为：

$$G_0(s)=G_{01}(s)G_{02}(s)=\frac{1.06}{(120s+1)(10s+1)}e^{-80s}$$

（2）PID 控制仿真

为进行系统 PID 控制的阶跃给定响应仿真，给出 MATLAB 程序 L5406a.m 如下。

```
% MATLAB PROGRAM L5406a.m
%
n1=[1.06];d1=conv([120 1],[10 1]);G1=tf(n1,d1);
tau=80;[np,dp]=pade(tau,2);Gp=tf(np,dp);
n21=[301.5 1.228];d21=[245.5 0];G21=tf(n21,d21);
n22=[9286 239.7 1.62];d22=[521.2 144.4 0];G22=tf(n22,d22);
Gc1=feedback(G1*G21,Gp);Gc2=feedback(G1*G22,Gp);
```

```
set(Gc1,'Td',tau);
step(Gc1);hold on
[y,t]=step(Gc1);
[sigma,tp,ts]=perf(1,y,t);
set(Gc2,'Td',tau);step(Gc2)
[y,t]=step(Gc2);
[sigma,tp,ts]=perf(1,y,t);
gtext('1  PI control'),
gtext('2  PID control')
```

程序运行后得到阶跃给定响应仿真曲线如图 5-86 所示，程序运行后还得到阶跃给定响应性能指标。

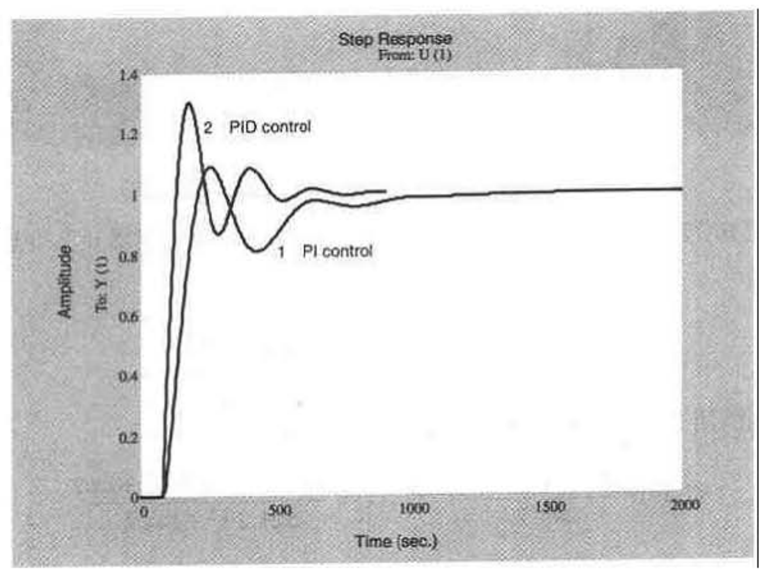


图 5-86 PI 与 PID 控制的阶跃给定响应曲线

PI 调节器仿真结果数据：

过程控制的超调量：*sigma* = 9.1289 %

过程控制的峰值时间：*tp* = 254.8883 s

过程控制的调节时间：*ts* = 289.8045 s

PID 调节器仿真结果数据：

过程控制的超调量：*sigma* = 30.2028 %

过程控制的峰值时间：*tp* = 175.4928 s

过程控制的调节时间：*ts* = 322.0134 s

两种调节器控制的过程各有所长，PI 调节器控制的超调量小，但是调节过程稍慢；PID 调节器控制的过程快一些，但是超调量大一些。

（3）史密斯预估器控制仿真

根据史密斯预估器控制原理与结构框图 5-83，给出实现其仿真的 MATLAB 程序

L5406b.m 如下。

```
% MATLAB PROGRAM L5406b.m
%
n1=[1.06];d1=conv([120 1],[10 1]);G1=tf(n1,d1);
tau=80;[np,dp]=pade(tau,2);Gp=tf(np,dp);
n21=[301.5 1.228];d21=[245.5 0];G21=tf(n21,d21);
n22=[9286 239.7 1.62];d22=[521.2 144.4 0];G22=tf(n22,d22);
Gc1=feedback(G1*G21,1);Gc2=feedback(G1*G22,1);
set(Gc1,'Td',tau);step(Gc1);hold on
set(Gc2,'Td',tau);step(Gc2)
gtext('1  PI control'),
gtext('2  PID control')
```

程序执行后,可得如图 5-87 所示的史密斯预估器 PI 与 PID 控制的系统阶跃给定响应曲线。首先看出，曲线显示史密斯预估器控制的作用完全实现，原来阶跃给定响应振荡的曲线，由预估器进行补偿控制后成为单调上升的过程；还可以看出，PI 与 PID 控制两者中，PID 控制的快速性能更好。可见史密斯预估器控制是一个有效的控制方法。

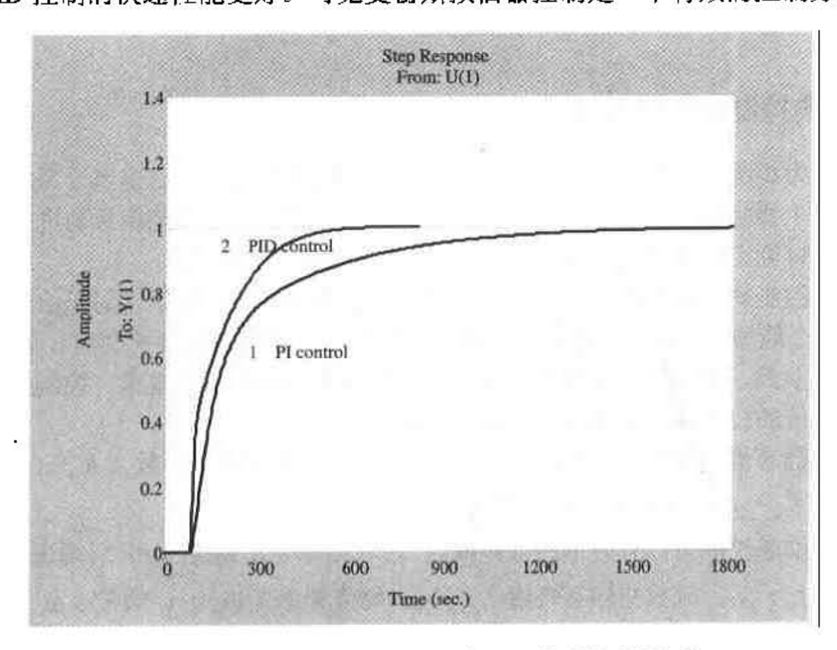


图 5-87　史密斯预估器 PI 与 PID 控制的阶跃响应

第 6 章　自动控制系统设计

由已知的系统结构、参数计算其性能指标并对系统仿真，这是系统分析问题。控制系统设计是与系统分析相反的问题：给定系统性能指标与给定的对象，设计能够完成特定任务的控制器。本章主要介绍控制系统的经典设计方法，本章共 4 节：6.1 节概述控制系统设计；6.2 节介绍控制系统 Bode 图设计法；6.3 节介绍控制系统根轨迹设计法；6.4 节介绍控制系统 PID 校正设计法。通过本章的学习，可以初步了解与掌握三种经典设计方法，并对控制系统进行所要求的设计。

6.1　控制系统设计概述

6.1.1　经典控制系统的设计

到目前为止所介绍的内容，主要着重于系统的计算与仿真。这是属于系统分析的范畴，解决的主要问题是：已知系统的结构与参数，计算系统的性能指标与进行相应的仿真。本节将讨论控制系统的设计问题，它是系统分析的逆向问题。

在经典控制理论中，所谓系统设计，就是在给定的性能指标下，对于给定的对象模型，确定一个能够完成给定任务的控制器（常称为校正器或者补偿控制器），即确定校正器的结构与参数。所谓给定任务即指系统满足的静态与动态性能要求。控制系统设计又叫做控制系统的校正或者控制系统的校正设计。

系统的静态性能指标有：稳态误差（e_{ss}），静态位置误差系数（K_p），静态速度误差系数（K_v），静态加速度误差系数（K_a）等。

系统的动态性能指标有时域的与频域的。时域指标为超调量（$\sigma\%$）、峰值时间（t_p）、调节时间（t_s）等；频域指标有峰值（A_m）、峰值频率（ω_m）、频带（ω_b）、剪切频率（ω_c）、模值稳定裕度（L_h）、相角稳定裕度（γ）等。

6.1.2　控制系统设计的几个概念

在经典控制系统里，主要数学模型是微分方程或传递函数。基于传递函数模型的有动态结构图、频率特性等。故控制系统设计方法有基于微分方程求解的分析微分方程特征根的根轨迹法，有基于频率特性的波德图法（Bode 图）。

按照校正器与给定受控对象联接的方式，有串联校正、反馈校正、前置校正与干扰补偿四种。

串联校正控制器的频域设计方法中，使用的校正器有超前校正器、滞后校正器以及超前滞后校正器等。它们分别使用在不同的场合，都很有效。

1．超前校正设计

超前校正设计是指利用校正器对数幅频曲线有正斜率（即幅频曲线的渐近线与横坐标夹角的正切值大于零）的区段及其相频曲线具有正相移（即相频曲线的相角值大于零）区段的系统校正设计。这种校正设计方法的突出特点是校正后系统的剪切频率比校正前的大，系统的快速性能得到提高。如果采用无源网络作校正器，会产生增益损失，现已被有源校正所代替。这种校正设计方法被要求稳定性好、超调小以及动态过程响应快的系统经常采用。

2．滞后校正设计

滞后校正设计是指利用校正器对数幅频特性曲线具有负斜率（即幅频曲线的渐近线与横坐标夹角的正切值小于零）的区段及其相频特性曲线具有负相移（即相频曲线的相角值小于零）区段的系统校正设计。这种校正设计方法的突出特点是校正后系统的剪切频率比校正前的小，系统的快速性能变差，但系统的稳定性能却得到提高。可见，在系统快速性要求不是很高，而稳定性与稳态精度要求很高的场合，滞后校正设计方法是很适合的。

3．滞后-超前校正设计

滞后—超前校正设计是指既有滞后校正作用又有超前校正作用的校正器设计。它既具有滞后校正高稳定性能、高精确度的长处，又具有超前校正响应快、超调小的优点。虽然这种校正所用装置或元器件要多些，会增加设备投资，但因其优良的性能，在要求高的设备中还是经常被采用。

6.1.3 控制系统的无源校正与有源校正

1．控制系统无源校正

早期控制系统的校正设计中所采用的是 R、C （电阻、电容）元件连接成的各种不同功能类型的校正器，这种校正器不带任何能源。故将这种采用 RC 无源网络作为校正器的校正称为控制系统的无源校正。RC 无源网络连接到系统里，有明显的负载效应，故使得校正效果的精度受到影响。

2．控制系统有源校正

现代控制系统校正设计中所采用的是集成运算放大器带不同反馈连接构成的各种功能类型的校正器，这些校正器是带电源的。故将这种采用集成运放的有源网络作为校正器的称为控制系统有源校正。现在工程实际里都毫不例外地大量而广泛地采用有源校正。

6.1.4 控制系统串联校正的基本思路

1．频域法校正的基本思路

频域法校正即是借助 Bode 图进行校正。当系统动态性能不能满足要求时，仅通过改变开环增益 K 来改善其动态性能，其结果往往是动态性能得到提高，而静态性能却降低了，以致于不能满足品质指标的要求。即改变开环增益 K 常常不能使静态、动态指标都满足要求。此时必须对系统进行校正。

为使系统既有较好的稳态性能，又有一定的稳定裕量，则可通过两个办法来实现。其一是，以满足系统稳态性能指标的开环增益为基础，对系统 Bode 图在剪切频率附近提供一个相位超前量，使之达到动态相角稳定裕量的要求，而保持低频部分不变。这个办法即是相位超前校正。其二是，仍以满足系统稳态性能指标的开环增益为基础，对系统 Bode 图保持低频段不变，将其中频与高频段的模值加以衰减，使之在中频段的特定点处，达到满足动态相角稳定裕量的要求。这个办法即是相位滞后校正。

2. 根轨迹法校正的基本思路

根轨迹法校正即是借助根轨迹曲线进行校正。系统的期望主导极点往往不在系统的根轨迹上。由根轨迹的理论，添加上开环零点或极点可以使根轨迹曲线形状改变。若期望主导极点在原根轨迹的左侧，则只要加上一对零、极点，使零点位置位于极点右侧。如果适当选择零、极点的位置，就能够使系统根轨迹通过期望主导极点 s_1，并且使主导极点在 s_1 点位置时的稳态增益满足要求。此即为相位超前校正。

系统的期望主导极点若在系统的根轨迹上，但是在该点的静特性不满足要求，即对应的系统开环增益 K 太小。单纯增大 K 值将会使系统阻尼比变小，甚至于使闭环特征根跑到复平面 S 的右半平面去。为了使闭环主导极点在原位置不动，并满足静态指标要求，则可以添加上一对偶极子，其极点在其零点的右侧。从而使系统原根轨迹形状基本不变，而在期望主导极点处的稳态增益得到加大。此乃是相位滞后校正。

6.2 控制系统波德图设计法

基于传递函数的频率特性，其图形化重要的可见形式之一就是 Bode 图。Bode 图法是频域设计中最常用的，这类方法主要分为超前校正设计、滞后校正设计以及两者的综合设计等三种方法。由于工程上三种方法中的任何一种都能够解决实际问题，所以本节将逐一介绍这三种常用校正设计方法的原理、步骤与方法并举例。

6.2.1 波德图超前校正设计

根据自动控制理论，超前校正环节的的两个转折频率应分设在未校正系统剪切频率的两侧。超前校正相频特性曲线具有正相移（即超前相位角），幅频特性曲线具有正斜率，校正后，系统的低频段 Bode 图不变，而剪切频率比原系统的大，即说明校正后系统快速性能得到提高。如果超前校正器采用 RC 无源网络，则有负载效应与增益损失之弊，故而工程上常采用由集成运算放大器构成的有源超前校正器。

6.2.1.1 波德图设计方法

根据自动控制的理论，利用波德图进行相位超前校正设计的步骤如下：

（1）求出满足题目给定误差系数性能指标的开环增益 K 值。

（2）根据求得的 K 值，画出校正前原系统的 Bode 图，并计算出模稳定裕量 G_m、$-\pi$ 穿越频率 ω_{cg}、相角稳定裕量 γ_1（即 P_m）、剪切频率 ω_{c1}（即 ω_{cp}）以检验性能指标是否

满足要求。若不满足要求，则执行下一步。

（3）为了满足相稳定裕量的要求，超前网络必须提供的相位超前角：

$$\phi_m=\gamma_0-\gamma_1+\left[\angle G(j\omega_{c1})H(j\omega_{c1})-\angle G(j\omega_{c2})H(j\omega_{c2})\right] \quad (6\text{-}1)$$

式中 γ_0——系统校正后期望的相角稳定裕量；

γ_1——系统校正前的相角稳定裕量；

ω_{c2}——系统校正后的剪切频率。

$\left[\angle G(j\omega_{c1})H(j\omega_{c1})-\angle G(j\omega_{c2})H(j\omega_{c2})\right]$加入了相位超前网络后，由于网络在中频及高频的增益提升特性，使剪切频率由ω_{c1}移到ω_{c2}所带来的原系统相位的滞后量，一般为$5^\circ\sim10^\circ$，故上式可写成

$$\phi_m=\gamma_0-\gamma_1+（5^\circ\sim10^\circ） \quad (6\text{-}2)$$

（4）根据以下关系式由ϕ_m求α值

$$\alpha=\frac{1-\sin\phi_m}{1+\sin\phi_m} \quad (6\text{-}3)$$

即 $\alpha<1$。

（5）确定校正后系统的剪切频率ω_{c2}

为了最大限度利用超前网络的相位超前量，ω_{c2}应与ω_m（产生ϕ_m角度的角频率为ω_m）相重合。在ω_m处$\left|\alpha G_c(j\omega)\right|$的提升量为$20\lg\sqrt{\alpha}$，所以$\omega_{c2}$应选在未校正系统的$L(\omega)=-20\lg\sqrt{\alpha}$处，当$\alpha$被补偿后（参见以下第（7）步），则$L(\omega)=20\lg\sqrt{\alpha}$。

（6）求校正补偿器的传递函数

$$\omega_{c2}=\omega_m=\frac{1}{\sqrt{\alpha}T}$$

或

$$T=\frac{1}{\sqrt{\alpha}\,\omega_m} \quad (6\text{-}4)$$

即可求得T值，于是有

$$G_c(s)=\frac{s+\frac{1}{T}}{s+\frac{1}{\alpha T}}=\alpha\cdot\frac{1+Ts}{1+\alpha Ts} \quad (6\text{-}5)$$

（7）在系统中把原放大器的增益增大$1/\alpha$倍，或者插入一个增益为$1/\alpha$的放大器，以补偿其增益损失。则有

$$G_c(s)=\frac{Ts+1}{\alpha Ts+1} \quad (6\text{-}6)$$

（8） 画出校正后的系统的Bode图，并校验系统性能指标。

（9） 使系统构成闭环，验证闭环系统的响应。

校正设计解题过程中，要特别注意系统开环增益的概念，它是指系统开环传递函数

中将分子分母 s 多项式最低阶系数换算为 1 后的总比例系数。开环增益或者定义为：系统开环传递函数中各典型环节的 s 零次方项（即 s^0 项如果有的话）的系数换算为 1 后的总比例系数。

【例 6-1】已知单位负反馈系统被控对象的传递函数为：

$$G_0(s)=K_0\frac{1}{s(0.1s+1)(0.001s+1)}$$

试用 Bode 图设计方法对系统进行超前串联校正设计，使之满足：

（1）在斜坡信号 $r(t)=v_0t$ 作用下，系统的稳态误差 $e_{ss}\leqslant 0.001v_0$；

（2）系统校正后，相角稳定裕度 γ 有：$43^\circ < \gamma < 48^\circ$。

【解】

根据利用波德图设计相位超前校正器的步骤，解题如下。

（1）求 K_0

根据自动控制理论与题意，本题给定系统为 I 型系统，在斜坡信号 $r(t)=v_0t$ 作用下，速度误差系数 $K_v=K=K_0$，式中 K 是系统的开环增益。系统的稳态误差

$$e_{ss}=\frac{v_0}{K_v}=\frac{v_0}{K}=\frac{v_0}{K_0}\leqslant 0.001v_0$$

$$K_v=K=K_0\geqslant 1000\ \text{s}^{-1}，取 K_0=1000\ \text{s}^{-1}$$

即被控对象的传递函数为：

$$G_0(s)=1000\frac{1}{s(0.1s+1)(0.001s+1)}$$

（2）做原系统的 Bode 图与阶跃响应曲线，检查是否满足题目要求

根据系统校正设计的步骤，首先检查原系统的频域性能指标是否满足题目要求，并观察其阶跃响应曲线形状，给出如下用 MATLAB 的程序 L6201a.m。

```
% MATLAB PROGRAM L6201a.m
%
k0=1000;
n1=1;d1=conv(conv([1 0],[0.1 1]),[0.001 1]);
[mag,phase,w]=bode(k0*n1,d1);
figure(1);margin(mag,phase,w);hold on
figure(2);s1=tf(k0*n1,d1);
sys=feedback(s1,1);step(sys)
```

程序运行后，可得到如图 6-1 所示的未经校正系统的 Bode 图及其性能指标，还有如图 6-2 所示的未校正系统的阶跃给定响应曲线。

由图 6-1 和图 6-2 可知系统的：模稳定裕量 $G_m\approx 0.1$dB；-π 穿越频率 $\omega_{cg}\approx 100.0\ \text{s}^{-1}$；

相稳定裕量 $P_m\approx 0.1$ deg；剪切频率 $\omega_{cp}\approx 99.5\ \text{s}^{-1}$

由计算的数据——相稳定裕量与模稳定裕量几乎为零，这样的系统是根本不能工作的。其阶跃响应曲线（图 6-2）剧烈振荡（虽然衰减），同样说明系统不能工作。

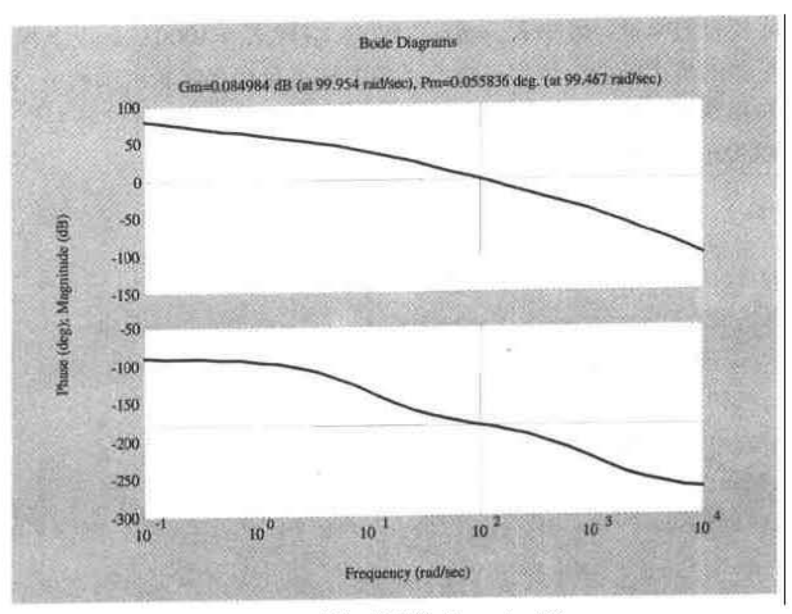


图 6-1 单闭环系统的 Bode 图

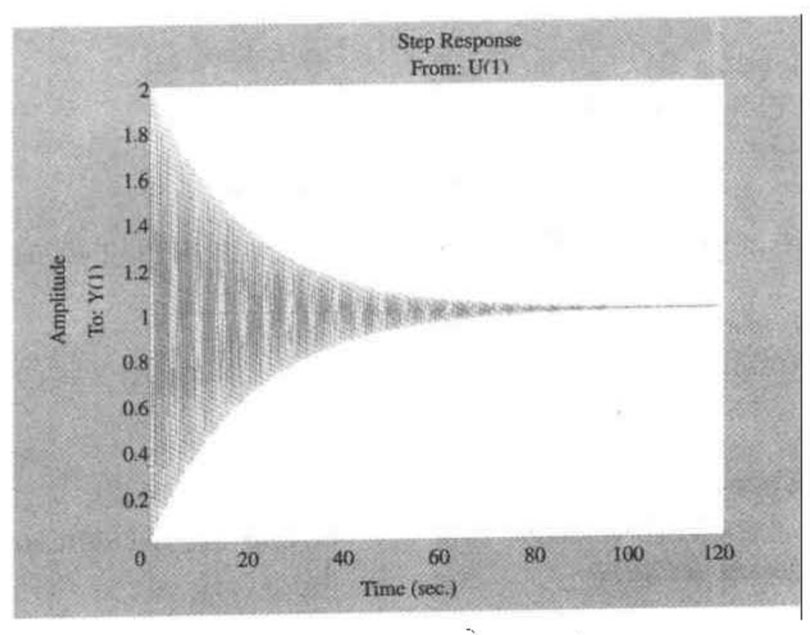


图 6-2 单闭环系统的单位阶跃响应

（3）求超前校正器的传递函数

由于原系统的固有缺陷，要想达到题意要求，必须对系统进行校正（设计补偿器）。根据要求的相稳定裕度$\gamma=45^\circ$ 并附加 5°，即取$\gamma=50^\circ$。

设超前校正器的传递函数为：

$$G_c(s)=\frac{Ts+1}{\alpha Ts+1}$$

根据超前校正的原理，可知 $K_v = K_0 \geqslant 1000s^{-1}$，取 $K_0 = 1000\ s^{-1}$。以下程序 L6201b.m 是求超前校正器的传递函数，程序可改为求超前校正器传递函数的函数 leadc( )。这个函数既可以已知相稳定裕度 γ ，也可以已知剪切频率，请参见本节后补充。

```
% MATLAB PROGRAM L6201b.m
%
k0=1000;
n1=1;d1=conv(conv([1 0],[0.1 1]),[0.001 1]);
sope=tf(k0*n1,d1);[mag,phase,w]=bode(sope);
gama=50;[mu,pu]=bode(sope,w);gama1=gama+5;
gam=gama1*pi/180;
alfa=(1-sin(gam))/(1+sin(gam));
adb=20*log10(mu);
am=10*log10(alfa);
ca=adb+am;
wc=spline(adb,w,am);
T=1/(wc*sqrt(alfa));
alfat=alfa*T;
Gc=tf([T 1],[alfat 1])
```

运行程序后即得校正器传递函数：

$$G(s)=\frac{Ts+1}{\alpha Ts+1}=\frac{0.01796s+1}{0.001786s+1}$$

如果执行以下调用函数 leadc( )的程序段，会得到相同的超前校正器传递函数：

```
k0=1000;
n1=1;d1=conv(conv([1 0],[0.1 1]),[0.001 1]);
sope=tf(k0*n1,d1);gama=50;wc=160;
[Gc]=leadc(1,sope,[gama]);
[Gc]=leadc(2,sope,[wc]);
```

（4）校验系统校正后系统是否满足题目要求

根据校正后系统的结构与参数，给出 以下 MATLAB 的程序 L6201c.m：

```
% MATLAB PROGRAM L6201c.m
%
k0=1000;
n1=1;d1=conv(conv([1 0],[0.1 1]),[0.001 1]);s1=tf(k0*n1,d1);
n2=[0.01796 1];d2=[0.001786 1];s2=tf(n2,d2);sope=s1*s2;
[mag,phase,w]=bode(sope);
margin(mag,phase,w)
```

程序运行后，可得校正后的 Bode 图，如图 6-3 所示。

由图 6-3 可知系统的： 模稳定裕量 G_m=17.693 dB；-π 穿越频率 $\omega_{cg} = 698.16\ s^{-1}$；

相稳定裕量 P_m=47.874 deg； 剪切频率 $\omega_{cp}=176.56\,\mathrm{s}^{-1}$

由程序算出的相角稳定裕量 P_m=47.874 deg，已经满足题目$43°<\gamma<48°$的要求。

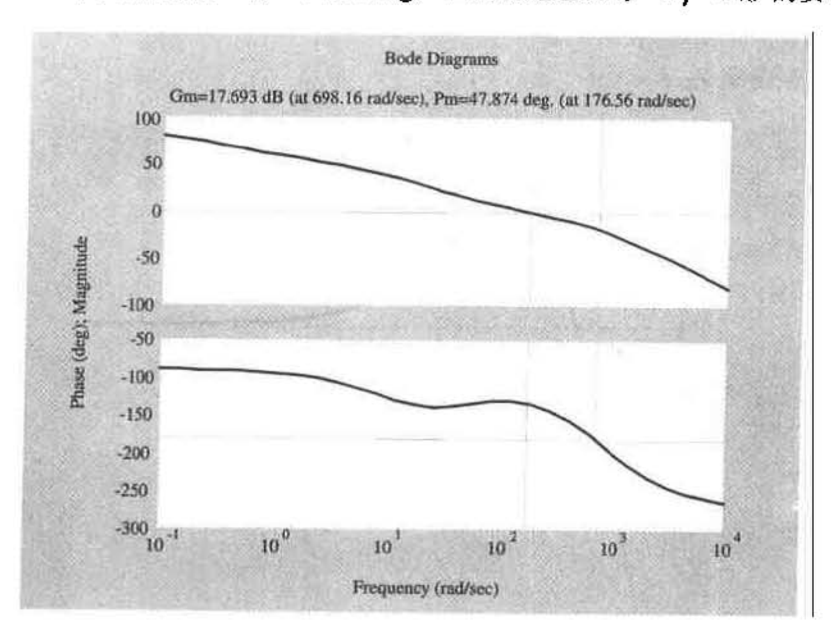


图 6-3 校正后系统的 Bode 图与频域性能指标

（5）计算系统校正后阶跃给定响应曲线及其性能指标

根据校正后系统的结构与参数，给出调用函数 perf.( )与 targ.( )求阶跃响应及其性能指标的 MATLAB 程序 L6201d.m 如下。

```
% MATLAB PROGRAM L6201d.m
%
global y t
k0=1000;
n1=1;d1=conv(conv([1 0],[0.1 1]),[0.001 1]);
s1=tf(k0*n1,d1);n2=[0.01796 1];d2=[0.001786 1];
s2=tf(n2,d2);sope=s1*s2;
sys=feedback(sope,1);
step(sys);[y,t]=step(sys);
[sigma ,tp,ts]=perf(1,y,t);
[mp,tp,b1,b2,n,pusi,T,f]=targ(y,t);
```

程序运行后可得校正后系统响应曲线，如图 6-4 所示。

还有校正后系统的阶跃响应品质指标：

超调量：*sigma* =25.7707 %

峰值时间：*tp* = 0.0158 s

调节时间：*ts*=0.0316 s

程控制的第一（正向）波峰值：*b*1 = 0.2593

过程控制的衰减比：n = 1
过程控制的衰减率：$pusi$ = 0
过程控制的衰减振荡周期：T = 0
过程控制的振荡频率：f = Inf

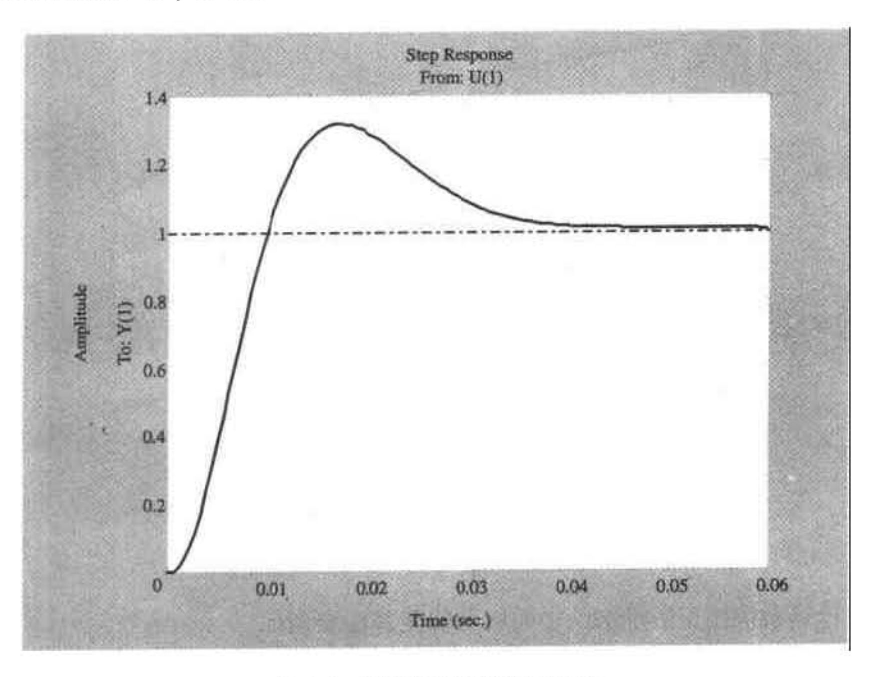


图 6-4 系统单位给定响应曲线

6.2.1.2 波德图设计的解析方法

设未经校正系统的传递函数为：$G_0(s)=K_0G_0'(s)$，式中 K_0 是满足系统误差系数性能指标的开环增益。

设串联校正补偿器的传递函数为：$G_c(s)=\dfrac{1+Ts}{1+\alpha Ts}$。串联校正后，系统则为 $G_0(s)G_c(s)$。其频率特性为

$$G_0(j\omega)G_c(j\omega)=A(j\omega)\,e^{j\varphi(j\omega)} \tag{6-7}$$

振幅频率特性在剪切频率 ω_c 处穿越横坐标，此时有

$$\begin{cases}|A(\omega_c)|=1\\ \varphi(\omega_c)=-180^\circ+P_m\end{cases} \tag{6-8}$$

于是有：

$$G_0(j\omega_c)G_c(j\omega_c)=M_Ge^{j\theta_G}\frac{j\omega_ct_z+1}{j\omega_ct_p+1}=1e^{j(-180+P_m)} \tag{6-9}$$

式中 $t_z=T,t_p=\alpha T$，M_G 与 θ_G 是 $G_0(j\omega_c)$ 在 ω_c 点的模值与相角。式(6-9)是一复数

方程，复数可以分为实数与虚数两部分。因此可以列写出两个方程，求解此二元一次联立方程，可得：

$$t_z=\frac{1+M_G\cos(P_m-\theta_G)}{-\omega_c M_G\sin(P_m-\theta_G)} \tag{6-10}$$

$$t_p=\frac{M_G+\cos(P_m-\theta_G)}{\omega_c\sin(P_m-\theta_G)} \tag{6-11}$$

即串联校正器 $G_c(s)=\dfrac{t_z s+1}{t_p s+1}$ ，能够唯一确定。

【例 6-2】续上题，试用 Bode 图设计的解析法对系统进行超前串联校正设计，使之满足：

（1）在斜坡信号 $r(t)=v_0 t$ 作用下，系统的稳态误差 $e_{ss}\leqslant 0.001v_0$；

（2）系统校正后，相角稳定裕度 γ 有：$\gamma\geqslant 48°$。

（3）剪切频率 $\omega_c\geqslant$ 170.0 s^{-1}（上题无此要求）。

【解】

（1）求串联校正补偿器的传递函数

$$G_c(s)=\frac{1+Ts}{1+\alpha Ts}$$

上题已经求得 K_0=1000 s-1。根据系统校正后，相角稳定裕度 $\gamma\geqslant$ 47° 的要求，取 γ=50°，还取 ω_c= 170.0 s-1。根据这些要求，给出 以下 MATLAB 程序 L6202a.m 来计算超前串联校正器。作为波德图设计解析法的程序，可改为求超前校正器传递函数的函数 leadc( )，这个函数解算的条件是已知相稳定裕度 γ 与剪切频率，请参见本节后“补充”。

```
% MATLAB PROGRAM L6202a.m
%
k0=1000;n1=1;d1=conv(conv([1 0],[0.1 1]),[0.001 1]);
sope=tf(k0*n1,d1);wc=170;gama=50;gama1=gama+5
num=sope.num{1};den=sope.den{1};
ngv=polyval(num,j*wc);
dgv=polyval(den,j*wc);
g=ngv/dgv;
thetag=angle(g);
thetag_d=thetag*180/pi;
mg=abs(g);
gama_rad=gama*pi/180;
z=(1+mg*cos(gama_rad-thetag))/(-wc*mg*sin(gama_rad-thetag));
p=(cos(gama_rad-thetag)+mg)/(wc*sin(gama_rad-thetag));
nc=[z,1];dc=[p,1]; Gc=tf(nc,dc)
```

运行程序后即得校正器传递函数：

$$G(s)=\frac{Ts+1}{\alpha Ts+1}=\frac{0.01647s+1}{0.0009388s+1}$$

如果执行以下调用函数 leadc( )的程序，会得到相同的校正器传递函数。

```
% MATLAB PROGRAM L6202b.m
%
k0=1000;
n1=1;d1=conv(conv([1 0],[0.1 1]),
[0.001 1]);sope=tf(k0*n1,d1);
wc=170;gama=50;[Gc]=leadc(3,sope,[gama,wc]);
```

（2）校验串联校正后系统是否满足题目要求

对于校正后的系统，求其 Bode 图及其频域性能指标，与要求的指标对照，即可知道校正器计算是否符合要求。为此给出以下 MATLAB 程序 L6202c.m 进行计算。

```
% MATLAB PROGRAM L6202c.m
%
k0=1000;n1=1;d1=conv(conv([1 0],[0.1 1]),[0.001 1]);
s1=tf(k0*n1,d1);s2=tf([0.01647 1],[0.0009388 1]);
sope=s1*s2;[mag,phase,w]=bode(sope);margin(mag,phase,w)
```

程序运行后，可得如图 6-5 所示的校正后的 Bode 图及其频域性能指标。

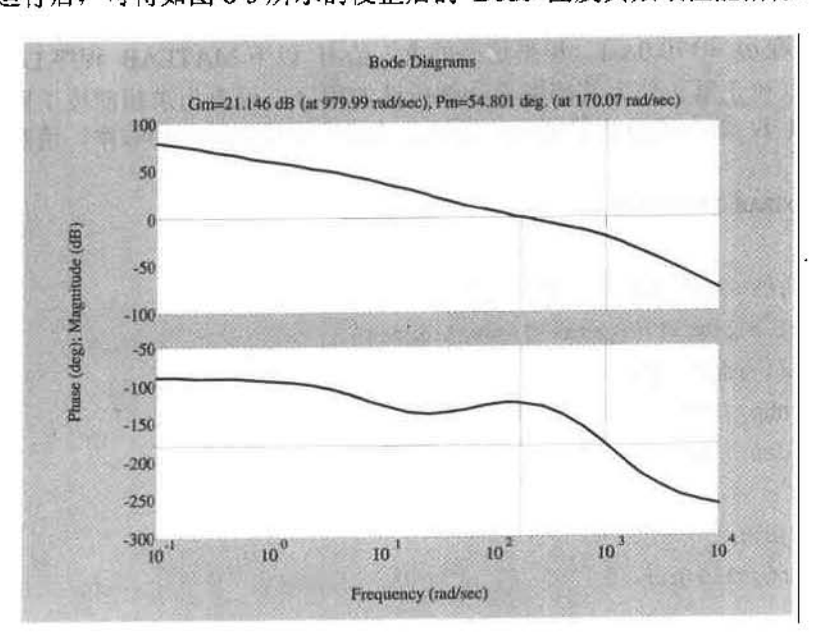


图 6-5　校正后系统的 Bode 图与频域性能指标

由图 6-5 可知系统的：模稳定裕度 G_m= 21.146 dB；$-\pi$ 穿越频率 $\omega_{cg}=979.99\ \text{s}^{-1}$；

相稳定裕度 P_m=54.801deg；剪切频率 $\omega_{cp}=170.07\ \text{s}^{-1}$

即校正后系统的相角稳定裕度 $\gamma=54.801°\geqslant 48°$，剪切频率 $\omega_c=170.07\ \text{s}^{-1}>170\ \text{s}^{-1}$，完

全满足要求。

用 Bode 图设计的解析方法计算系统校正后阶跃响应曲线及其性能指标与【例 6-1】计算结果相近。

6.2.2 波德图滞后校正设计

根据自动控制理论，滞后校正环节的的两个转折频率应比未校正系统剪切频率小得多，或者是校正后系统剪切频率的（$\frac{1}{10}\sim\frac{1}{2}$）。滞后校正相频特性曲线具有负相移（即滞后相位角），幅频特性曲线具有负斜率。校正后，系统的低频段 Bode 图不变，而剪切频率比原系统的小，即说明校正后系统快速性能变差。这意味着滞后校正是以牺牲系统的快速性来换取系统稳定性的。如果滞后校正器采用 RC 无源网络，虽无增益损失，却有负载效应之弊，故而工程上也采用由集成运算放大器构成的有源滞后校正器。

根据自动控制理论，利用波德图设计相位滞后校正器的步骤如下：

（1）求出满足题目给定误差系数性能指标的开环增益 K 值。

（2）根据求得的 K 值，画出校正前原系统的 Bode 图，并计算出模稳定裕量 G_m、$-\pi$ 穿越频率 ω_{cg}、相角稳定裕量 γ_1（即 P_m）、剪切频率 ω_{c1}（即 ω_{cp}）以检验性能指标是否满足要求。若不满足要求，则执行下一步。

（3） 根据题目要求，选择一新的剪切频率 ω_{c2}，使得在 $\omega=\omega_{c2}$ 处原系统的相位滞后角为：

$$\phi(\omega_{c2})=-180^\circ+\gamma_0+(5^\circ\sim12^\circ) \tag{6-12}$$

式中 γ_0——系统校正后期望的相角稳定裕量；

$(5^\circ\sim12^\circ)$——相位滞后网络在 $\omega=\omega_{c2}$ 处所引起的相位滞后量。

（4）求校正网络的 β 值（滞后校正器的传递函数为：$G_c(s)=\frac{1+Ts}{1+\beta Ts}$，$\beta>1$）。在 $\omega=\omega_{c2}$ 处有：

$$20\log\left(\frac{1}{\beta}\right)+L(\omega_{c2})=0\ \text{dB}$$

即

$$20\log\beta=L(\omega_{c2})$$

则

$$\beta=1/10^{-L(\omega_{c2})/20} \tag{6-13}$$

（5）选择校正网络的零点

从理论上讲，$1/T$ 距离 ω_{c2} 越远，相位滞后网络的滞后特性对系统影响越小，故 $1/T$ 选得越小越好。但是因为当 $1/T$ 距 ω_{c2} 一定远时，$1/T$ 的减小对 ω_{c2} 点的相位滞后量的变化影响不大。

$$\frac{1}{T}=\left(\frac{1}{10}\sim\frac{1}{2}\right)\omega_{c2} \tag{6-14}$$

则有
$$G_c(s)=\frac{1+Ts}{1+\beta Ts} \tag{6-15}$$

（6）画出校正后的系统的 Bode 图，并校验系统性能指标。

（7） 使系统构成闭环，验证闭环系统的响应。

【例 6-3】已知单位负反馈系统被控对象的传递函数为：

$$G_0(s)=K_0\frac{1}{s(0.1s+1)(0.2s+1)}$$

试用 Bode 图设计方法对系统进行滞后串联校正设计，使之满足：

（1）在单位斜坡信号 $r(t)=t$ 作用下，系统的速度误差系数 $K_v \geqslant 30\ s^{-1}$；

（2）系统校正后剪切频率 $\omega_c \geqslant 2.3\ s^{-1}$ ；

（3）系统校正后相角稳定裕度γ有：$\gamma > 40°$。

【解】

（1）求 K_0

根据自动控制理论与题意，本题给定系统为 I 型系统，单位斜坡响应的速度误差系数 $K_v=K=K_0$，式中 K 是系统的开环增益。即有：

$$K_v=K=K_0 \geqslant 30\ s^{-1}，取 K_0=30\ s^{-1}$$

则被控对象的传递函数为：

$$G_0(s)=30\frac{1}{s(0.1s+1)(0.2s+1)}$$

（2）做原系统的 Bode 图与阶跃响应曲线，检查是否满足题目要求

根据系统校正设计的步骤，首先检查原系统的频域性能指标是否满足题目要求，并观察其阶跃响应曲线形状。为此，给出如下用 MATLAB 的函数命令 bode ( )、step ( ) 列写的仿真程序为 L6203 a.m：

```
% MATLAB PROGRAM L6203a.m
%
k0=30;n1=1;d1=conv(conv([1 0],[0.1 1]),[0.2 1]);
s1=tf(k0*n1,d1);[mag,phase,w]=bode(s1);
figure(1); margin(mag,phase,w);hold on
figure(2);sys=feedback(s1,1);step(sys)
```

程序运行后，可得未校正系统的 Bode 图与频域性能图 6-6，还有如图 6-7 所示的未校正系统阶跃响应曲线。

由计算数据可知未校正系统的频域性能：

模稳定裕量 G_m= -6.0 dB； -π 穿越频率 $\omega_{cg}=7.1\ s^{-1}$；

相稳定裕量 P_m= -17.1 deg； 剪切频率 $\omega_{cp}=9.8\ s^{-1}$

由计算的结果——相稳定裕量与模稳定裕量均为负值可知，这样的系统是根本不能工作的。这也可从发散振荡的阶跃响应曲线（图 6-7）看到，系统必需修校正。

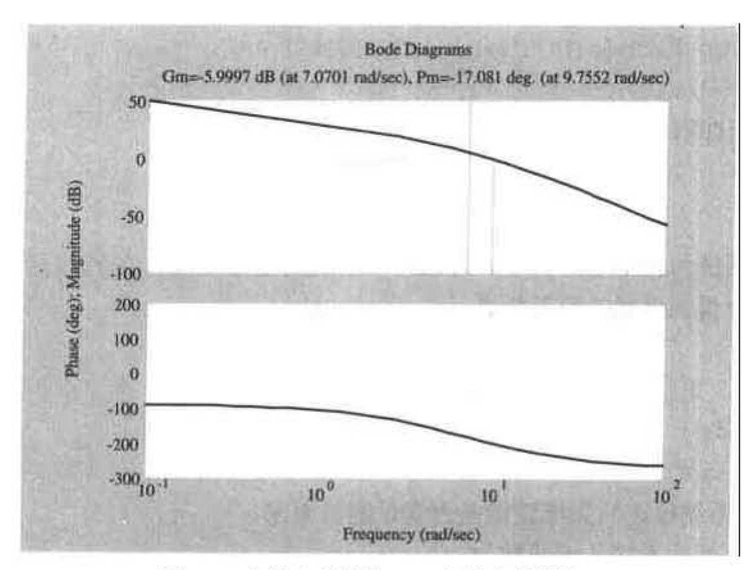


图 6-6　未校正系统的 Bode 图与频域性能

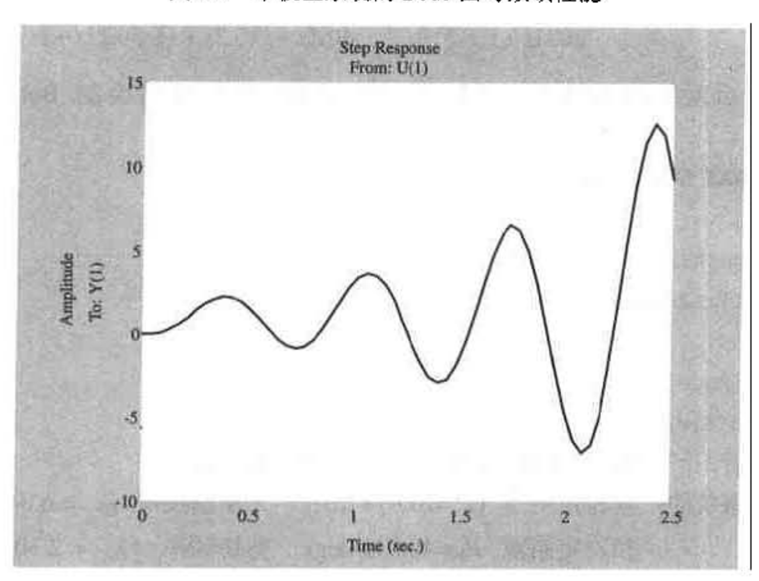


图 6-7　未校正系统的单位阶跃响应

（3）求滞后校正器的传递函数

取校正后系统的剪切频率 $\omega_{c2}=2.3\text{s}^{-1}$。根据滞后校正的原理及题意，给出求滞后校正器传递函数的 MATLAB 程序 L6203b.m 如下。程序可以改为求滞后校正器传递函数的函数 lagc( )。如果已知系统的开环传递函数与要校正后的剪切频率或者相角稳定裕度，求串联滞后校正器，都可调用这个函数。请参见本节后“补充”。

```
% MATLAB PROGRAM  L6203b.m
%
wc=2.3;k0=30;n1=1;d1=conv(conv([1 0],[0.1 1]),[0.2 1]);
na=polyval(k0*n1,j*wc);da=polyval(d1,j*wc);
```

```
g=na/da;g1=abs(g);h=20*log10(g1);beta=10^(h/20);
T=1/(0.1*wc);bt=beta*T;Gc=tf([T 1],[bt 1])
```

程序运行后即得校正补偿器传递函数：

$$G_c(s)=\frac{1+Ts}{1+\beta Ts}=\frac{4.348s+1}{50.21s+1}$$

如果已知系统的相角稳定裕度，可以用以下调用函数 lagc( )的程序段求校正补偿器传递函数，也会得到满意的校正效果。

```
k0=30;n1=1;
d1=conv(conv([1 0],[0.1 1]),[0.2 1]);
sope=tf(k0*n1,d1);
gama=46;[Gc]=lagc(1,sope,[gama]);
```

（4）校验系统校正后频域性能是否满足题目要求

包含有校正器的系统传递函数为：

$$G_0(s)\,G_c(s)=\frac{4.348s+1}{50.21s+1}\,G_0(s)=30\frac{1}{s(0.1s+1)(0.2s+1)}\,\frac{4.348s+1}{50.21s+1}$$

根据校正后系统的结构与参数，用 MATLAB 函数编写绘制 Bode 图的程序 L6203c.m：

```
% MATLAB PROGRAM L6203c.m
%
k0=30;n1=1;d1=conv(conv([1 0],[0.1 1]),[0.2 1]);
s1=tf(k0*n1,d1);n2=[4.348 1];d2=[50.21 1];s2=tf(n2,d2);
sope=s1*s2;
[mag,phase,w]=bode(sope);
margin(mag,phase,w)
```

程序运行后，可得校正后系统的 Bode 图，如图 6-8 所示。

由图可知系统的：模稳定裕量 G_m=14.854 dB；　-π 穿越频率　ω_{cg} = 6.84 s^{-1}；

相稳定裕量 P_m=46.845 deg；　剪切频率　ω_{cp} = 2.3021 s^{-1}

由程序计算出的相角稳定裕量 P_m=46.845 deg，已经满足题目对系统校正后相角稳定裕度 $\bar{a}>40°$ 的要求。

（5）计算系统校正后阶跃响应曲线及其性能指标

根据校正后系统的结构与参数，用 MATLAB 命令函数，编写求阶跃响应及其性能指标的程序 L6203d.m。

```
% ATLAB PROGRAM L6203d.m
%
global y t
k0=30;n1=1;d1=conv(conv([1 0],[0.1 1]),[0.2 1]);
s1=tf(k0*n1,d1);n2=[4.348 1];d2=[50.21 1];s2=tf(n2,d2);
sope=s1*s2;sys=feedback(sope,1);
```

```
step(sys)
[y,t]=step(sys);
[sigma ,tp,ts]=perf(1,y,t);
```

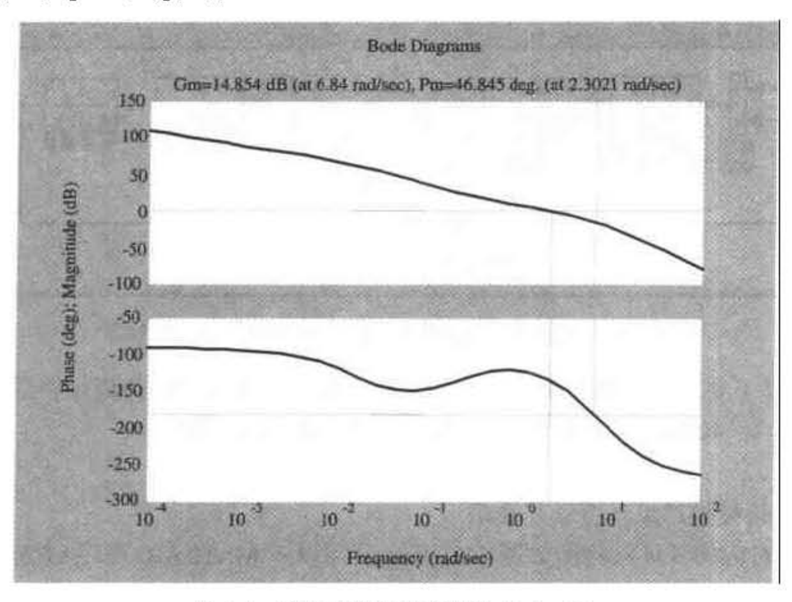


图 6-8 校正后单闭环系统的 Bode 图

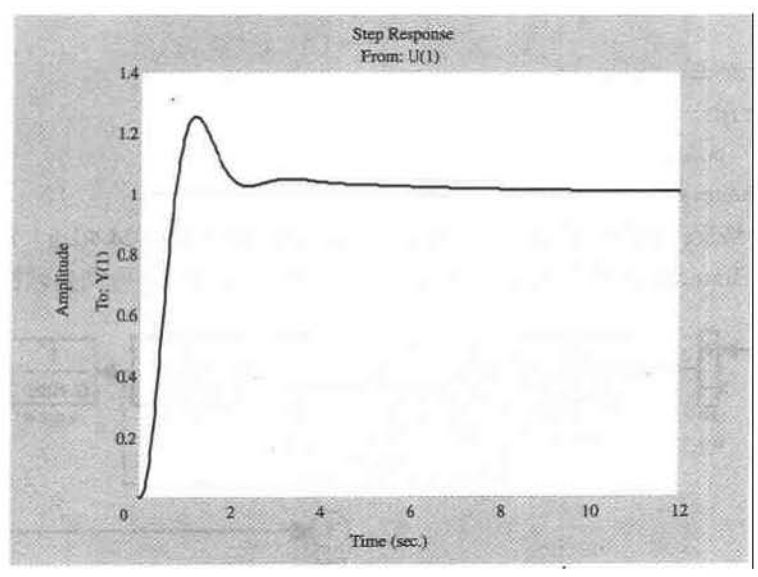


图 6-9 校正后系统的单位阶跃响应

程序运行后有校正后系统的单位阶跃响应曲线，如图 6-9 所示，校正后系统的阶跃响应品质指标是：

超调量：*sigma* = 24.7725 %

峰值时间：*tp* = 1.2019 s

调节时间：*ts* = 1.9729 s

【例 6-4】已知晶闸管-直流电机单闭环调速系统（V-M 系统）的 SIMULINK 动态结构图如图 6-10 所示。

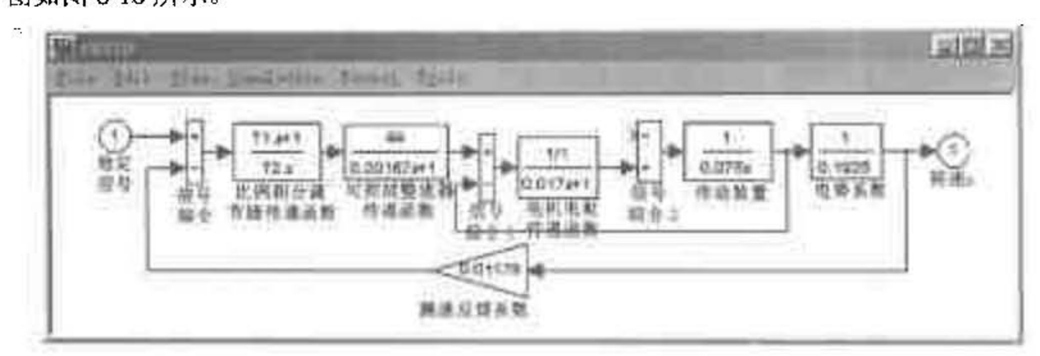


图 6-10　转速单闭环调速系统的 SIMULINK 动态结构图

试按照以下要求设计速度调节器（滞后校正器）：校正后相角稳定裕量γ ≥50°；剪切频率 ω_c ≥ 40.0 s^{-1}；超调量$\sigma\%$ ≤30 %；调节时间 ts≤0.1 s。

【解】

（1）检验被控对象是否满足要求

图 6-10 对应的开环结构图如图 6-11 所示，用以下 MATLAB 程序 L6204a.m 来计算其频域性能指标。

```
% MATLAB PROGRAM L6204a.m
%
[a,b,c,d]=linmod2('t6211');
sope=ss(a,b,c,d);
[mag,phase,w]=bode(sope);
margin(mag,phase,w)
```

程序运行结果：模稳定裕量 G_m=-13.3dB；相稳定裕量 P_m=-24.4deg。这样的系统必须校正。不用 linmod( )函数还可以用传递函数的方法，可以同样得到必须校正的结论。

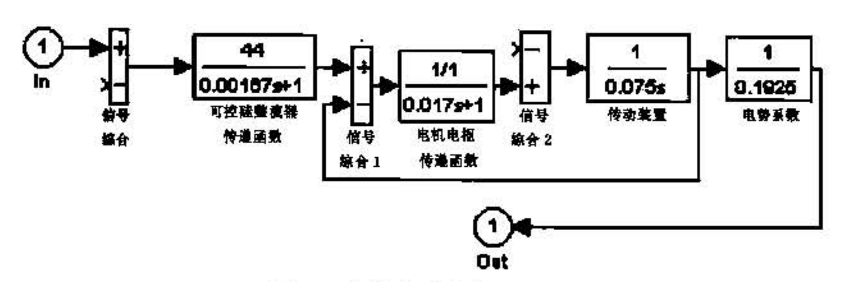


图 6-11　被控对象构成的结构图

（2）求滞后校正器的传递函数

给出以下调用函数 lagc( )的 MATLAB 程序 L6204b.m 来求滞后校正器的传递函数。

```
% MATLAB PROGRAM L6204b.m
%
[a,b,c,d]=linmod2('t6211');
```

```
s1=ss(a,b,c,d);sope=tf(s1);
gama=50;wc=40;
[Gc]=lagc(1,sope,[gama]);
```

执行程序后即得滞后校正器的传递函数为：

$$G_c(s)=\frac{1+Ts}{1+\beta Ts}=\frac{0.204s+1}{0.1298s+1}$$

（3）检验校正后系统的频域性能指标

将校正器Gc(s)串入系统前向通道，对系统开环绘制 Bode 图并计算频域性能指标，可以用以下调用函数 lagc( )的程序 L6204c.m 来实现。

```
% MATLAB PROGRAM L6204c.m
%
n1=1;d1=[0.017 1];s1=tf(n1,d1);
n2=1;d2=[0.075 0];s2=tf(n2,d2);
s12=feedback(s1*s2,1);s3=tf([44],[0.00167 1]);
s4=tf([1],[0.1925]);s5=tf([0.01178],[1]);
s6=tf([0.204 1],[0.1298 1]);
sope=s12*s3*s4*s5*s6;
[mag,phase,w]=bode(sope);
margin(mag,phase,w)
```

程序运行后可得系统校正后如图 6-12 所示的 Bode 图及其频域性能指标。

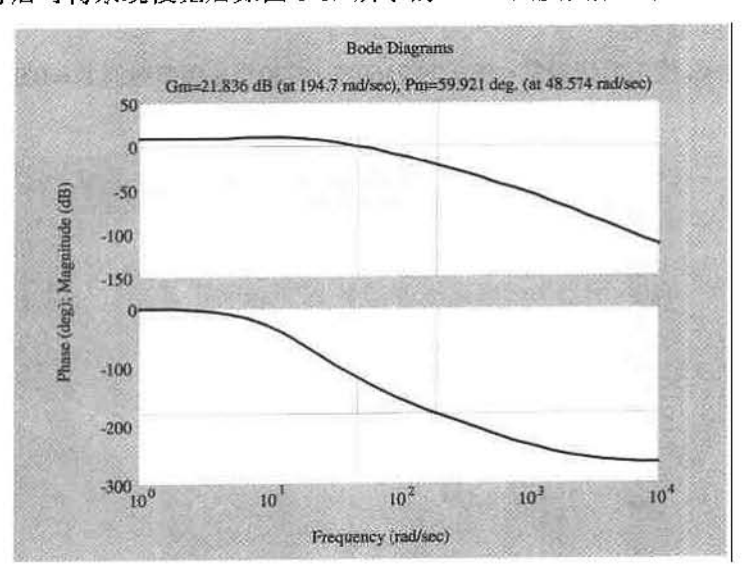


图 6-12 校正后单闭环系统的 **Bode** 图

由图可知系统的：模稳定裕量 G_m=21.836 dB；-π 穿越频率 $\omega_{cg}=194.7\ s^{-1}$；

相稳定裕量 P_m=59.921 deg； 剪切频率 $\omega_{cp}=48.574\ s^{-1}$

按照题意，相稳定裕量 γ =59.921°＞50° ； 剪切频率 ω_c =48.574s^{-1}＞40.0 s^{-1}，已满足

题目要求。

（4）检验校正后系统的时域性能指标

已经求出的频域性能指标是合格的。再将系统的反馈通道闭合，求其闭环系统的单位阶跃给定响应的性能指标。给出以下程序 **L6204d.m** 实现其计算。

```
% MATLAB PROGRAM L6204d.m
%
n1=1;d1=[0.017 1];s1=tf(n1,d1);
n2=1;d2=[0.075 0];s2=tf(n2,d2);
s12=feedback(s1*s2,1);
s3=tf([44],[0.00167 1]);
s4=tf([1],[0.1925]);
s5=tf([0.01178],[1]);
s6=tf([0.204 1],[0.1298 1]);
sope=s12*s3*s4*s6;
sys=feedback(sope,s5);step(sys)
[y,t]=step(sys);
[sigma ,tp,ts]=perf(1,y,t)
```

程序运行后可得校正后闭环系统的单位阶跃响应曲线如图 6-13 所示，并有：

sigma = 29.7160 %

tp = 0.0572 s

ts = 0.0883 s（5% 误差带）

核对题目要求，校正后超调量 $\sigma\%$ = 29.7160 %（≤30 %）；调节时间 ts = 0.0883s（≤0.1s），指标完全合格。

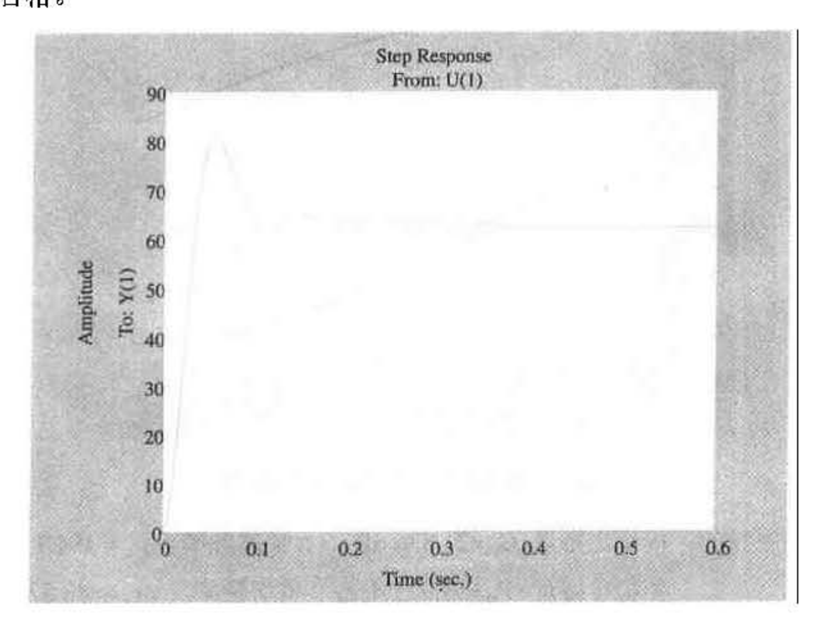


图 6-13　校正后闭环系统的单位阶跃响应

原书缺页

（6）画出校正后的系统的 Bode 图，并校验系统性能指标。

（7）使系统构成闭环，验证闭环系统的响应。

【例 6-5】已知单位负反馈系统被控对象的传递函数为：

$$G_0(s)=K_0\frac{1}{s(s+1)(s+2)}$$

试用 Bode 图设计方法对系统进行超前-滞后串联校正设计，使之满足：

（1）在单位斜坡信号 $r(t)=t$ 作用下，系统的速度误差系数 $K_v=10\ s^{-1}$；

（2）系统校正后剪切频率 $\omega_c \geqslant 1.5\ s^{-1}$ ；

（3）系统校正后相角稳定裕度γ有：$\gamma \geqslant 45°$ 。

（4）计算校正后系统时域性能指标：$\sigma\% \leqslant 25\%$，$t_p \leqslant 2$ s，$t_s \leqslant 6$ s。

【解】

（1）求 K_0

已知单位负反馈系统被控对象为Ⅰ型系统。根据自动控制理论，单位斜坡响应的速度误差系数 $K_v=K=10\ s^{-1}$，式中 K 是系统开环增益。根据速度误差系数的定义，有：

$$K_v=\lim_{s\to 0}s\cdot G_0(s)=\lim_{s\to 0}s\cdot\frac{K_0}{s(s+1)(s+2)}=10$$

则得 $K_0=20\ s^{-1}$。即被控对象的传递函数为：

$$G_0(s)=\frac{20}{s(s+1)(s+2)}$$

（2）作原系统的 bode 图与阶跃响应曲线，检查是否满足题目要求

根据系统校正设计的步骤，首先检查原系统的频域性能指标是否满足题目要求，并观察其阶跃响应曲线形状。为此，给出 MATLAB 程序 L6205a.m 如下。

```
% MATLAB PROGRAM L6205a.m
%
k0=20;n1=1;d1=conv(conv([1 0],[1 1]),[1 2]);
sope=tf(k0*n1,d1);[mag,phase,w]=bode(sope);figure(1);
margin(mag,phase,w);hold on
figure(2);sys=feedback(sope,1);step(sys)
```

程序运行后，可得未校正系统的 Bode 图，如图 6-14 所示，还有如图 6-15 所示的未校正系统阶跃响应曲线。

由计算数据可知未校正系统的频域性能：

模稳定裕量 G_m = -10.424 dB；　-π 穿越频率 $\omega_{cg}=1.414\ s^{-1}$；

相稳定裕量 P_m = -28.033 deg；　剪切频率 $\omega_{cp}=2.4243\ s^{-1}$

由计算的数据—相稳定裕量与模稳定裕量均为负值可知，这样的系统根本不能工作。这也可从发散振荡的阶跃响应曲线（图 6-15）看到，系统必须校正。

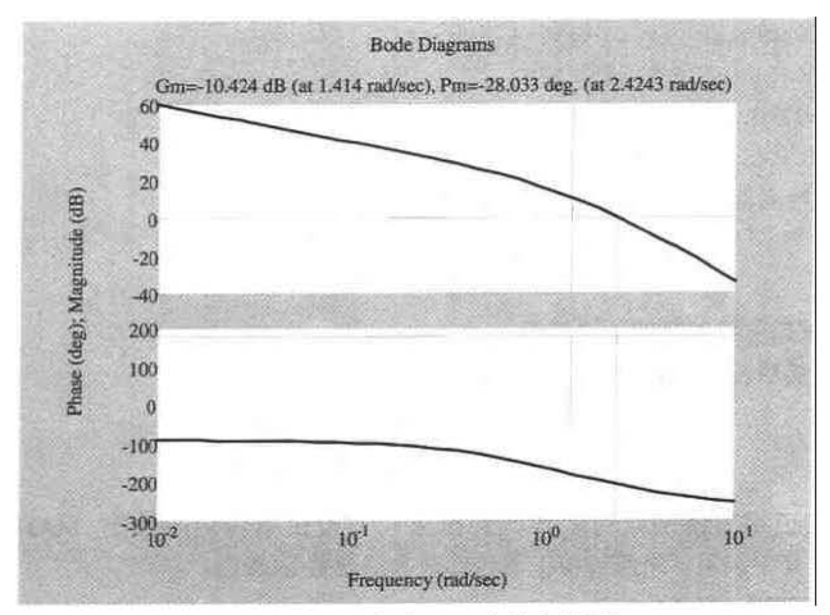


图 6-14 未校正系统的 Bode 图与频域指标

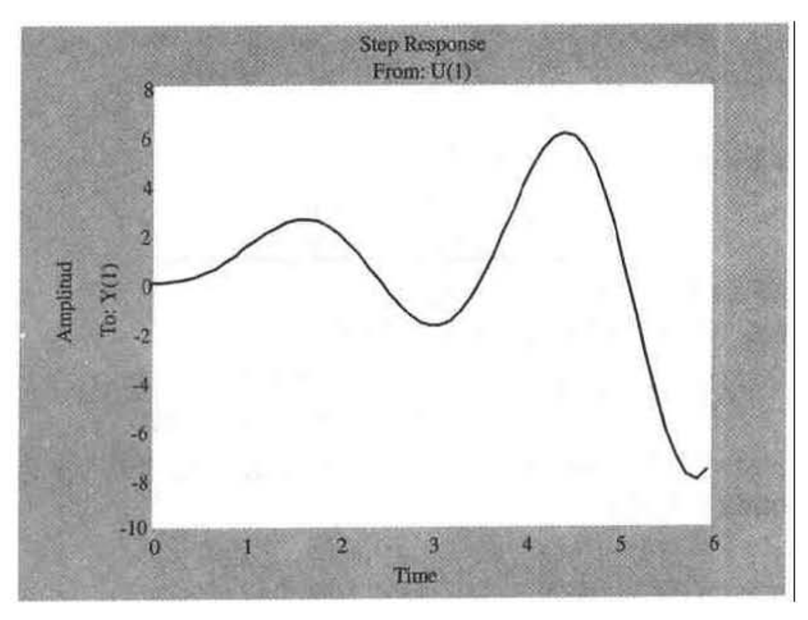


图 6-15 未校正系统的阶跃响应曲线

（3）求滞后校正器的传递函数

$$G_{c1}(s)=\frac{1+Ts}{1+\beta Ts}$$

根据题目要求，取校正后系统的剪切频率 $\omega_c=1.5\ \mathrm{s}^{-1}$， $\beta=9.5$。并编写求滞后校正器传递函数的 MATLAB 程序 L6205b.m。

```
% MATLAB PROGRAM  L6205b.m
%
```

```
wc=1.5;k0=20;n1=1;
d1=conv(conv([1 0],[1 1]),[1 2]);
beta=9.5;T=1/(0.1*wc);
betat=beta*T;Gc1=tf([T 1],[betat 1])
```

程序运行后即得滞后校正补偿器传递函数：

$$G_{c1}(s)=\frac{1+Ts}{1+\beta Ts}=\frac{6.667s+1}{63.33s+1}$$

（4）求超前校正器的传递函数

串联有滞后校正器的系统传递函数为：

$$G_0(s)\,G_{c1}(s)=\frac{20}{s(s+1)(s+2)}\cdot\frac{6.667s+1}{63.33s+1}$$

根据校正后系统的结构参数，给出求超前校正器传递函数的 MATLAB 程序 L6205c.m，其中调用了求超前校正器传递函数的函数 leadc( )。

```
%MATLAB PROGRAM L6205c.m
%
n1=conv([0 20],[6.667 1]);
d1=conv(conv(conv([1 0],[1 1]),[1 2]),[63.33 1]);
sope=tf(n1,d1);wc=1.5;[Gc]=leadc(2,sope,[wc])
```

程序运行后，即得超前校正器的传递函数：

$$G_{c2}(s)=\frac{1+Ts}{1+\alpha Ts}=\frac{2.13s+1}{0.2087s+1}$$

（5）校验系统校正后频域性能是否满足题目要求

包含滞后校正器与超前校正器的系统传递函数为：

$$G_0(s)\,G_{c1}(s)G_{c2}(s)=\frac{20}{s(s+1)(s+2)}\cdot\frac{6.667s+1}{63.33s+1}\cdot\frac{2.13s+1}{0.2087s+1}$$

给出如下 MATLAB 程序 L6205d.m，校验系统校正后频域性能是否满足题目要求。

```
% MATLAB PROGRAM L6205d.m
%
n1=20;d1=conv(conv([1 0],[1 1]),[1 2]);s1=tf(n1,d1);
s2=tf([6.667 1],[63.33 1]);s3=tf([2.13 1],[0.2087 1]);
sope=s1*s2*s3;[mag,phase,w]=bode(sope);
margin(mag,phase,w)
```

程序运行后，可得校正后的 Bode 图，如图 6-16 所示。

由图可知系统的：模稳定裕量 G_m=12.023 dB；-π 穿越频率 $\omega_{cg}=3.4829\,s^{-1}$；

相稳定裕量 P_m=47.119 deg；　剪切频率 $\omega_{cp}=1.4832\,s^{-1}$

由程序计算出的相角稳定裕量 $P_m=47.119$ deg，已经满足题目系统校正后 $\gamma>45°$ 的要求；剪切频率 $\omega_c=1.4832s^{-1}\approx1.5\,s^{-1}$ 也已经满足题目要求。

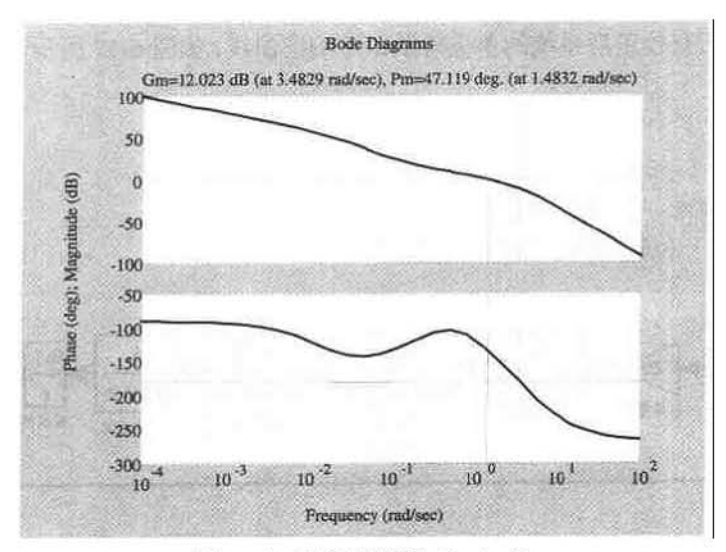


图 6-16 校正后系统的 Bode 图

（6）计算系统校正后阶跃给定响应曲线及其性能指标

根据校正后系统的结构与参数，给出如下求阶跃响应及其性能指标的 MATLAB 程序 L6205e.m。

```
% MATLAB PROGRAM L6205e.m
%
n1=20;d1=conv(conv([1 0],[1 1]),[1 2]);s1=tf(n1,d1);
s2=tf([6.667 1],[63.33 1]);s3=tf([2.13 1],[0.2087 1]);
sope=s1*s2*s3;sys=feedback(sope,1);step(sys)
[y,t]=step(sys);
[sigma]=perf(1,y,t)
```

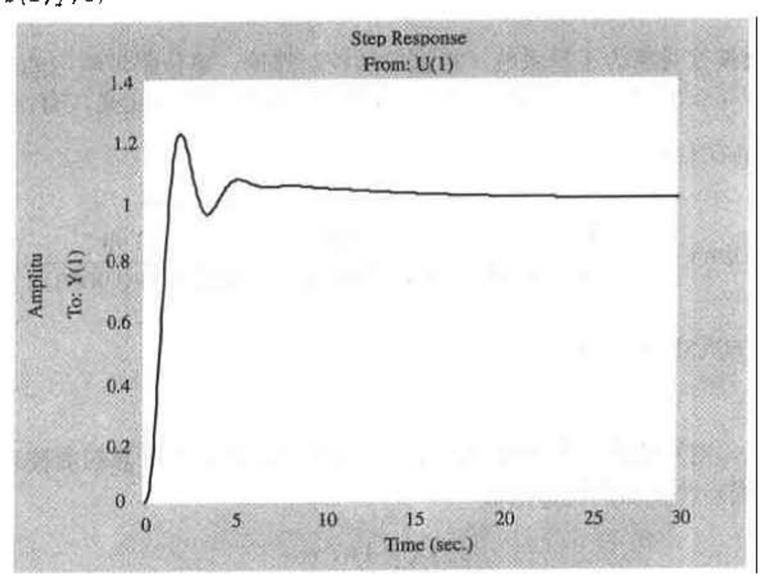


图 6-17 校正后单闭环系统的单位阶跃响应

程序运行后有校正后系统的单位阶跃给定响应曲线，如图 6-17 所示，其性能指标为：

超调量：*sigma* = 21.8863 %

峰值时间：*tp* = 1.9151 s

调节时间：*ts* = 5.9580 s

核对题目要求，时域性能指标均合格。

【例 6-6】已知一位置随动系统如图 6-18 所示。

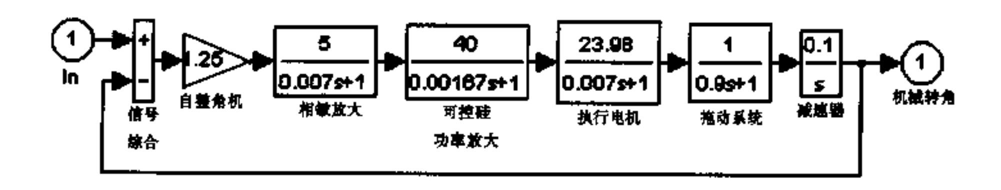


图 6-18 位置随动系统

试用 Bode 图设计方法对系统进行滞后-超前串联校正设计，使之满足：

（1）在单位斜坡信号 $r(t)=t$ 作用下，系统的速度误差系数 K_v=600 s^{-1}。

（2）系统校正后剪切频率 $\omega_c \geqslant 50$ s^{-1} 。

（3）系统校正后相角稳定裕度γ有：$\gamma \geqslant 40°$ 。

（4）系统阶跃给定响应超调量$\sigma\% \leqslant 35$ %。

【解】

（1）求 K_0

由图 6-18 可以得到随动系统前向通道的传递函数：

$$G_0(s)=1.25\frac{5}{0.007s+1}\cdot\frac{40}{0.00167s+1}\cdot\frac{23.98}{0.007s+1}\cdot\frac{1}{0.9s+1}\cdot\frac{0.1}{s}$$

已知系统被控对象为 I 型系统。根据自动控制理论，单位斜坡响应的速度误差系数 $K_v=K$=600 s^{-1}，式中 K 是系统开环增益。根据速度误差系数的定义，有：

$$K_v=\lim_{s\to 0}s\cdot G_0(s)$$

$$=\lim_{s\to 0}s\cdot K_0\cdot\frac{0.125}{s}\cdot\frac{5}{0.007s+1}\cdot\frac{40}{0.00167s+1}\cdot\frac{23.98}{0.0063s^2+0.9063s+1}$$

$$=605.53*K_0=600$$

则 $K_0\approx 1$。

根据题意，对系统进行复合校正，先计算滞后校正器，再计算超前校正器。

（2）求滞后校正器的传递函数

$$G_{c1}(s)=\frac{1+Ts}{1+\beta Ts}$$

根据题目要求，取校正后系统的剪切频率 $\omega_c=50\text{s}^{-1}$，$\beta=10$。给出求滞后校正器传递函数的 MATLAB 程序 L6206a.m 如下。

```
% MATLAB PROGRAM  L6206a.m
%
wc=50;beta=10;T=1/(0.1*wc);
betat=beta*T;
Gc=tf([T 1],[betat 1])
```

程序运行后即滞后校正器传递函数：

$$G_{c1}(s)=\frac{1+Ts}{1+\beta Ts}=\frac{0.2s+1}{2s+1}$$

（3）求超前校正器的传递

根据校正后系统的结构与参数，用 MATLAB 函数编写求超前校正器的传递函数的程序 L6206b.m，其中调用了超前校正的 MATLAB 函数 leadc( )。

```
% MATLAB PROGRAM L6206b.m
%
G1=tf(1.25,1);G2=tf(5,[0.007 1]);
G3=tf(40,[0.00167 1]);d3=conv([0.007 1],[0.9 1]);
G4=tf(23.98,d3);G5=tf(0.1,[1 0]);G6=tf([0.2 1],[2 1]);
sope=G1*G2*G3*G4*G5*G6;wc=50;
[Gc]=leadc(2,sope,[wc]);
```

程序运行后，可得超前校正器的传递函数：

$$G_{c2}(s)=\frac{1+Ts}{1+\alpha Ts}=\frac{0.8416s+1}{0.0004753s+1}$$

（4）校验系统校正后频域性能是否满足题目要求

对于前向通道串联滞后校正器与超前校正器的系统开环，校验校正后频域性能是否满足题目要求，给出 MATLAB 程序 L6206c.m 来进行计算。

```
% MATLAB PROGRAM L6206c.m
%
G1=tf(1.25,1);G2=tf(5,[0.007 1]);
G3=tf(40,[0.00167 1]);d3=conv([0.007 1],[0.9 1]);
G4=tf(23.98,d3);G5=tf(0.1,[1 0]);G6=tf([0.2 1],[2 1]);
G7=tf([0.8416 1],[0.0004753 1]);
G17=G1*G2*G3*G4*G5*G6*G7;[mag,phase,w]=bode(G17);
margin(mag,phase,w)
```

程序运行后，可得校正后如图 6-19 所示的 Bode 图及其频域性能指标。

由图可知系统的：模稳定裕量 G_m=9.8637 dB；$-\pi$ 穿越频率 $\omega_{cg}=107.01\ \text{s}^{-1}$；

相稳定裕量 P_m=39.697 deg；剪切频率 $\omega_{cp}=49.843\ \text{s}^{-1}$

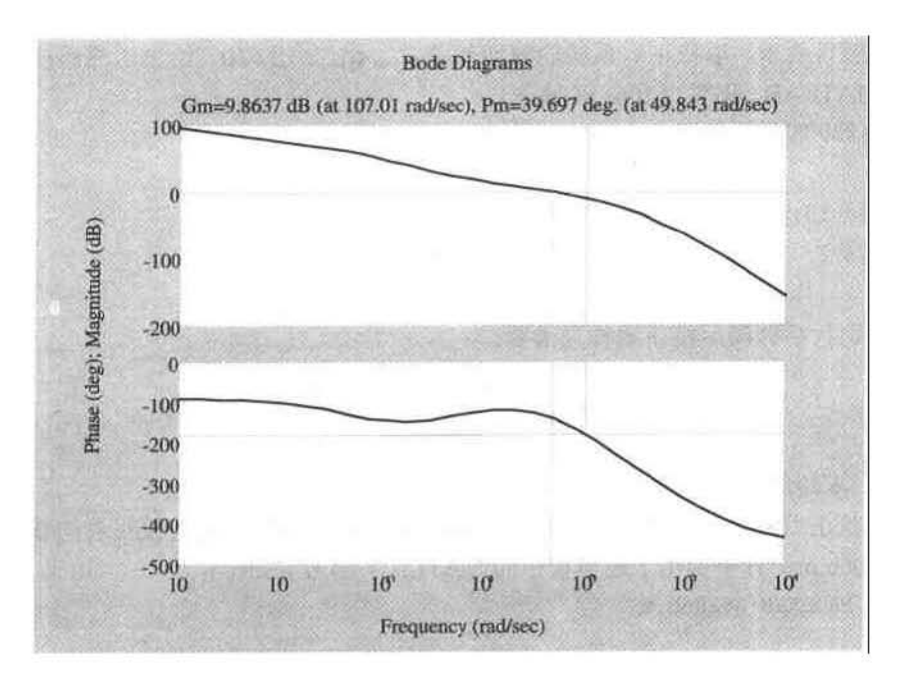


图 6-19　校正后系统的 Bode 图

校核题目要求，系统校正后剪切频率 ω_c =49.843 s^{-1}≈50 s^{-1} ；系统校正后相角稳定裕度 γ=39.697°≈40° ，系统校正与仿真合格。

（5）计算系统校正后阶跃给定响应曲线及其性能指标

给出如下程序 L6206d.m，用来计算系统校正后阶跃给定响应曲线及其性能指标。

```
% MATLAB PROGRAM L6206d.m
%
G1=tf(1.25,1);G2=tf(5,[0.007 1]);
G3=tf(40,[0.00167 1]);d3=conv([0.007 1],[0.9 1]);
G4=tf(23.98,d3);G5=tf(0.1,[1 0]);
G6=tf([0.2 1],[2 1]);
G7=tf([0.8416 1],[0.0004753 1]);
G17=G1*G2*G3*G4*G5*G6*G7;
Gc=feedback(G17,1);
step(Gc)[y,t]=step(Gc);
[sigma ,tp,ts]=perf(1,y,t);
```

程序运行后有校正后系统的单位阶跃给定响应曲线，如图 6-20 所示，其性能指标为：

超调量：*sigma* = 34.6130 %

峰值时间：*tp* = 0.0573 s

调节时间：*ts* = 0.1707 s

核对题目要求，系统阶跃给定响应超调量 $\sigma\%$ =34.6130 %（≤35 %），合格。

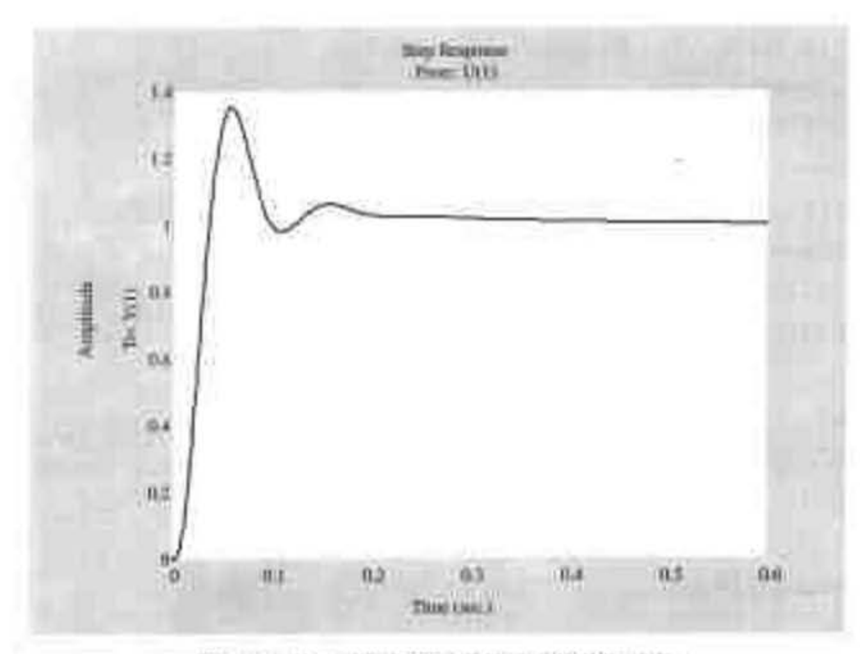


图 6-20 校正后位置随动系统的单位阶跃响应

附录：

【MATLAB FUNCTION PROGRAM 5】求系统串联超前校正器传递函数的函数 **leadc( )**

leadc( )函数的调用格式为：

[*Gc*]=**leadc**(*key*,**sope**,*var*)

其中 sope 是系统的开环传递函数。当 *key*=1 时，为 *var*=*gama*，是根据要求校正后的相角稳定裕度求校正器；当 *key*=2 时，为 *var*=*wc*，则是根据要求校正后的剪切频率求校正器。当 *key*=3 时，为 *var*(1)=*gama*，*uar*(2)=*wc*，则是既根据要求校正后的相角裕度，同时又根据要求校正后的剪切频率求校正器。若已知系统的开环传递函数与要求校正后的相角稳定裕度或剪切频率，求系统串联超前校正器传递函数时，调用此函数是最佳选择。

```
function [Gc]=leadc(key,sope,var)
% MATLAB FUNCTION PROGRAM leadc.m
%
if key==1
  gama=var (1) ;gama1=gama+5;
  [mag,phase,w]=bode(sope);
  [mu,pu]=bode(sope,w);
  gam=gama1*pi/180;
  alpha=(1-sin(gam))/(1+sin(gam));
  adb=20*log10(mu);
  am=10*log10(alfa);
  wc=spline(adb,w,am);
```

```
   T=1/(wc*sqrt(alfa));    % 根据式（6-3）求 T
   alphat=alpha*T;
   Gc=tf([T 1],[alphat 1]);
 elseif key==2
   wc=var (1)
   num=sope.num{1};den=sope.den{1};
   na=polyval(num,j*wc);
   da=polyval(den,j*wc);
   g=na/da;
   g1=abs(g);
   h=20*log10(g1);
   a=10^(h/10);
   wm=wc;
   T=1/(wm*(a)^(1/2));
   alphat=a*T ;
   Gc=tf([T 1],[alphat 1]);
elseif key==3
   gama=vars(1);wc=vars(2);gama1=gama+5;
   num=sope.num{1};den=sope.den{1};
   ngv=polyval(num,j*wc);
   dgv=polyval(den,j*wc);
   g=ngv/dgv;
   thetag=angle(g);
   thetag_d=thetag*180/pi;
   mg=abs(g);
   gama_rad=gama1*pi/180;
   z=(1+mg*cos(gama_rad-thetag))/(-wc*mg*sin(gama_rad-thetag));
   p=(cos(gama_rad-thetag)+mg)/(wc*sin(gama_rad-thetag));
   nc=[z,1];dc=[p,1];
   Gc=tf(nc,dc)
end
```

【MATLAB FUNCTION PROGRAM 6】求系统串联滞后校正器传递函数的函数 lagc( )

lagc( )函数的调用格式为：

[*Gc*]=lagc(*key*,sope,*var*)

其中 sope 是系统的开环传递函数，当 *key*=1 时，为 *var*=*gama*，是根据要求校正后的相角稳定裕度计算滞后校正器；当 *key*=2 时，为 *var*=*wc*，则是根据要求校正后的剪切频率计算校正器。若已知系统的开环传递函数与要求校正后的相角稳定裕度或剪切频率，求系统串联滞后校正器传递函数时，就可以调用此函数。

```
function [Gc]=lagc(key,sope,vars)
% MATLAB FUNCTION PROGRAM lagc.m
%
if key==1
    gama=vars(1);
    gama1=gama+5;
    [mu,pu,w]=bode(sope);
    wc=spline(pu,w',(gama1-180))
elseif key==2
   wc=vars(1);
end
   num=sope.num{1};den=sope.den{1};
   na=polyval(num,j*wc);
   da=polyval(den,j*wc);
   g=na/da;g1=abs(g);
   h=20*log10(g1);
   beta=10^(h/20);
   T=10/wc;
   betat=beta*T;
   Gc=tf([T 1],[betat 1])
```

6.3 控制系统的根轨迹设计法

6.3.1 根轨迹超前校正设计

本章开始已经介绍根轨迹相位超前校正。若期望主导极点 s_1 在原根轨迹的左侧，则只要加上一对零、极点，使零点位置位于极点右侧。当适当选择零、极点的位置，就能够使系统根轨迹通过期望主导极点 s_1，并且使主导极点在 s_1 点位置时的稳态增益满足要求。本节先介绍基于根轨迹图形的超前校正设计几何法，然后再介绍由几何图形关系演变公式的解析法。两种方法介绍后都有例题及其解算过程供参考。

6.3.1.1 根轨迹超前校正设计的几何法

设未校正系统的传递函数为 $G_0(s)$，超前校正器的传递函数为：

$$G_c(s)=K_c\cdot G_c'(s)=K_c\frac{s+a}{s+b} \tag{6-23}$$

则校正后系统的开环传递函数为：

$$G_{ope}(s)=G_0(s)\cdot G_c(s)=G_0(s)\cdot K_c\frac{s+a}{s+b} \tag{6-24}$$

用根轨迹超前校正几何方法的设计步骤如下：

（1）根据要求的动态品质指标，确定闭环主导极点 s_1 的位置。s_1 的位置确定后，该点在复平面的相角为：

$$\varphi=\angle(s_1) \tag{6-25}$$

（2）计算使根轨迹通过主导极点 s_1 所需的补偿角 φ_c 。补偿角 φ_c 的计算公式为：

$$\varphi_c=180^\circ-\angle(s_1) \tag{6-26}$$

（3）按图 6-21 确定 $G_c(s)$ 的零极点，可使其附加增益最小，其方法是：

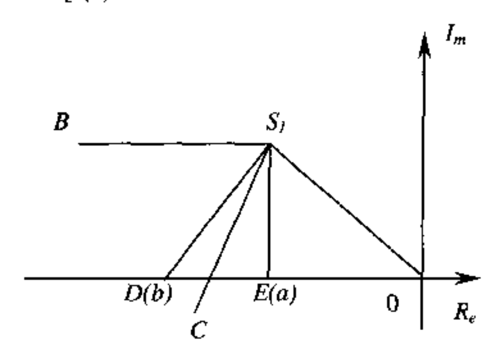


图 6-21　确定根轨迹超前校正器零极点

①过 s_1 点作水平线 s_1B，则 $\angle Bs_10=\varphi$ 。

②作 $\angle Bs_10$ 的角平分线 s_1C。

③在线 s_1C 两边作 $\angle Ds_1C=\angle Es_1C=\varphi_c/2$ 。线 s_1D、s_1E 与负实轴的交点坐标分别为 b、a，则可确定超前校正器的极、零点：

$$G_c'(s)=\frac{s+a}{s+b}$$

（4）由根轨迹幅值条件即

$$|G_0(s_1)\cdot G_c(s_1)|=1 \tag{6-27}$$

计算校正器的增益 K_c 。

根据图 6-21，由平面几何的基本知识，以下几个关系式在对系统用根轨迹超前校正几何方法设计 MATLAB 程序时是必须要用到的。

令 $\angle s_1Do=\theta p$, 则
$$\theta_p=\frac{\varphi-\varphi_c}{2} \tag{6-28}$$

令 $\angle s_1Eo=\theta z$, 则
$$\theta_z=\frac{\varphi+\varphi_c}{2} \tag{6-29}$$

令 $-b=p_c$, 则
$$p_c=-b=\mathrm{Re}(s_1)-\frac{\mathrm{Im}(s_1)}{\tan\theta_p} \tag{6-30}$$

令 $-a=Z_c$, 则
$$z_c=-a=\mathrm{Re}(s_1)-\frac{\mathrm{Im}(s_1)}{\tan\theta_z} \tag{6-31}$$

【例 6-7】已知单位负反馈系统被控对象的传递函数为：

$$G_0(s)=\frac{1}{s(s+2)}$$

试用根轨迹几何方法对系统进行超前串联校正设计，使之满足：

（1）阶跃响应的超调量$\sigma\%\leqslant 20\%$；

（2）阶跃响应的调节时间 $t_s\leqslant 1.2$ s。

【解】

（1）确定希望极点在 S 复平面上的位置

根据题意，要求超调量$\sigma\%\leqslant 20\%$，用以下 MATLAB 语句求ς。

```
sigma=0.2;
zeta=((log(1/sigma))^2/((pi)^2+(log(1/sigma))^2))^(1/2)
```

可以求得ς为：

zeta=0.4559，即 $\varsigma\geqslant 0.4559$，取$\varsigma=0.5$，由$t_s(5\%)=\dfrac{3}{\varsigma\omega_n}=1$，可得$\omega_n=6$ rad/s。

然后根据根轨迹法则，给出以下 MATLAB 语句求在 S 复平面上希望极点的位置。

```
zeta=0.5;wn=6;
p=[1 2*zeta*wn wn*wn];
roots(p)
```

运行以上 MATLAB 语句，求得根：

```
ans =
  -3.0000 + 5.1962i
  -3.0000 - 5.1962i
```

即希望极点的位置为 $s_{1,2}=-3.0000\pm 5.1962i$。

（1）求校正补偿器的传递函数

$$G_c(s)=K_c\frac{s+a}{s+b}$$

根据关系式（6-23）~（6-31），给出以下程序 L6301a.m 计算校正器的传递函数。程序作适当修改，明确已知条件与待求参量，可以成为用根轨迹几何法求超前校正器传递函数的函数 rtloc( )，使用这个函数需要已知未校正系统的开环传递函数与希望极点在 S 复平面上的位置，请参见本节后“附录”。

```
% MATLAB PROGRAM L6301a.m
%
s1=-3.0000+5.1962i;
ng=1;dg=[1 2 0];
ngv=polyval(ng,s1);
dgv=polyval(dg,s1);g=ngv/dgv;
zeta=angle(g);
    if zeta>0;
```

```
        phic=pi-zeta;
    end;
    if zeta<0;
        phic=-zeta;
    end;
phi=angle(s1);
zetaz=(phi+phic)/2;
zetap=(phi-phic)/2;
zc=real(s1)-imag(s1)/tan(zetaz);
pc=real(s1)-imag(s1)/tan(zetap);
nc=[1 -zc];dc=[1 -pc];
nv=polyval(nc,s1);dv=polyval(dc,s1);
kv=nv/dv;kc=abs(1/(g*kv));
    if zeta<0;
        kc=-kc;
    end;
kc
Gc=tf(nc,dc)
```

程序运行后，可得校正器的 K_c 与传递函数：

```
kc =
  49.1667
Transfer function:
s + 3.875
---------
s + 9.292
```

运行以下调用 rtloc( )函数的程序段可以得到相同的结果。

```
s1=-3.0000+5.1962i;
n1=1;d1=[1 2 0];sope=tf(n1,d1);
[Gc]=rtloc(1,sope,[s1]);
```

即校正器传递函数为：

$$G_c(s)=K_c\frac{s+a}{s+b}=49.1667\frac{s+3.8745}{s+9.2915}$$

那么校正后系统传递函数为:

$$G_0(s)G_c(s)=\frac{1}{s(s+2)}\cdot K_c\frac{s+a}{s+b}=49.1667\frac{1}{s(s+2)}\cdot\frac{s+3.8745}{s+9.2915}$$

（3）校检校正器计算是否合要求

根据校正后的结构与参数，给出调用函数 perf( )的程序 L6301b.m 来计算系统时域性能指标。

```
% MATLAB PROGRAM L6301b.m
%
global y t
s1=tf(49.1667,[1 2 0]);
s2=tf([1 3.8745],[1 9.2915]);
sope=s1*s2;
sys=feedback(sope,1);
step(sys); [y,t]=step(sys); [sigma,tp,ts]=perf(1,y,t);
```

运行该程序，可得系统单位给定阶跃响应曲线，如图 6-22 所示，并有

sigma = 25.6531 %

tp = 0.5521 s

ts = 0.8650 s

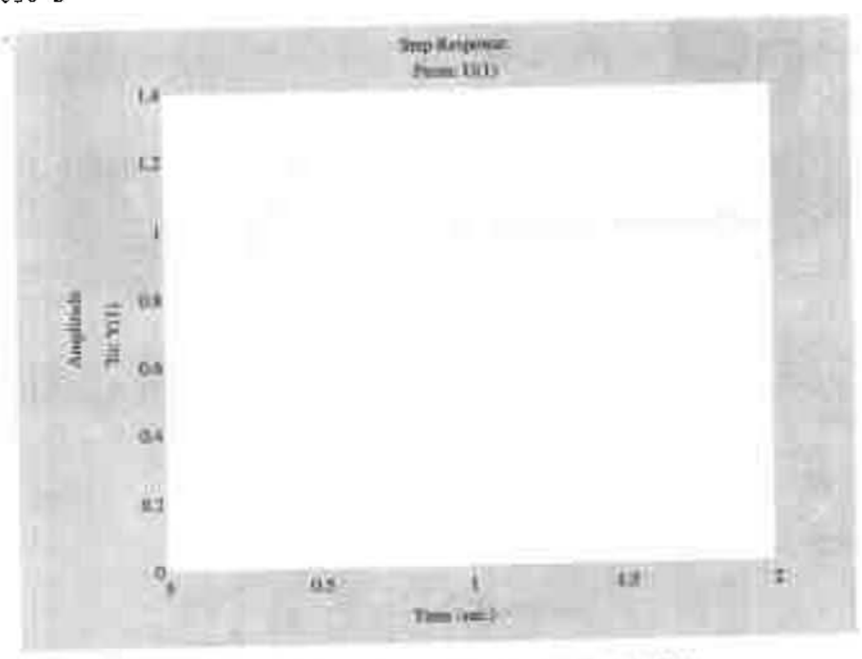


图 6-22 按根轨迹法校正后系统的阶跃响应曲线

即校正后系统的阶跃响应品质指标是：调节时间 $ts = 0.8650$ s。由以上数据看出，$ts < 1$ s，调节时间已满足要求；超调量 $\sigma\% = 25.6531\%$，没有达到要求。

（4）重新选择 ς、ω_n，再计算校正器的传递函数并校验是否满足要求

重选 $\varsigma = 0.6$，可得 $\omega_n = 5$ rad/s。求得校正补偿器的传递函数为：

$$G_c(s)=K_c\frac{s+a}{s+b}=33.4924\frac{s+3.0776}{s+8.1231}$$

即有校正后系统的传递函数

$$G_0(s)G_c(s)=\frac{1}{s(s+2)}\cdot K_c\frac{s+a}{s+b}=33.4924\frac{1}{s(s+2)}\cdot\frac{s+3.0776}{s+8.1231}$$

校检校正器计算是否正确，按以上计算，仍然要校验阶跃响应的品质指标。经计算，得改变参数后系统的阶跃响应曲线，如图6-23所示。并有

sigma =17.5754 %

tp = 0.6994 s

ts = 1.0675 s

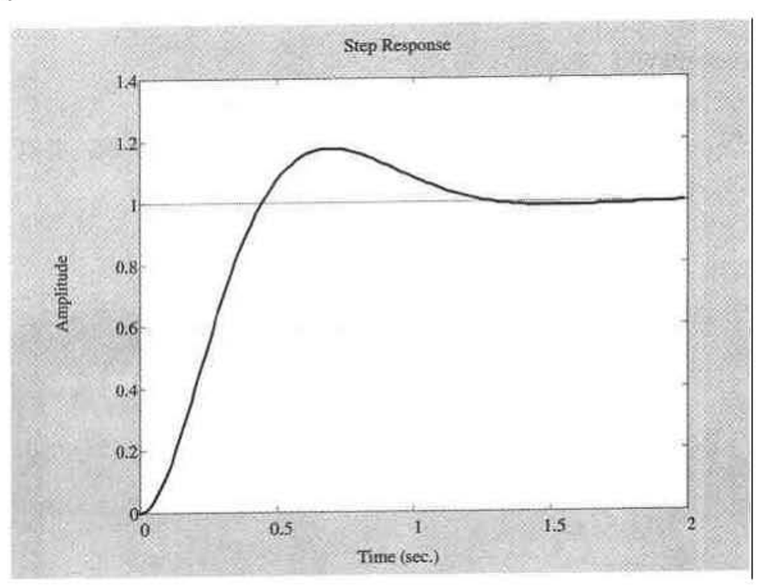


图6-23　改变参数后系统的阶跃响应曲线

即改变参数后的校正系统，超调量$\sigma\%=17.5754\%<20\%$，调节时间$ts=1.0675s\approx1s$。由以上数据可知，阶跃响应品质指标超调量与调节时间已满足题目要求。

6.3.1.2　根轨迹超前校正设计的解析法

设未校正系统传递函数为$G_0(s)$，校正器的传递函数为：

$$G_c(s)=K_c\frac{t_z s+1}{t_p s+1} \tag{6-32}$$

根据性能指标要求，确定K_c与期望极点s_1在s平面上的位置。则校正后系统的根轨迹为：

$$G_0(s_1)G_c(s_1)=K_c\frac{t_z s_1+1}{t_p s_1+1}M_G e^{j\theta G}=1e^{j\pi} \tag{6-33}$$

式中

$$G_0(s_1)=M_G e^{j\theta G}$$

K_c已经确定，需要对于式（6-33）求解t_z与t_p以确定$G_c(s)$。期望极点s_1在s平面上的位置也已确定，对于复变量s_1可用模幅式表示：

$$s_1=M_S e^{j\theta S} \tag{6-34}$$

式(6-34)带入式（6-33）可得：

$$M_s e^{j\theta_s} t_z + 1 = \left[\frac{1e^{j\pi}}{K_c M_G e^{j\theta_G}}\right]\left(M_s e^{j\theta_s} t_p + 1\right) \tag{6-35}$$

式（6-35）是一个复数方程，对其实部与虚部可以分别得到含有未知数t_z与t_p的两个方程，联立求解即得：

$$t_z = \frac{\sin\theta_s - K_c M_G \sin(\theta_G - \theta_s)}{K_c M_G M_s \sin\theta_G} \tag{6-36}$$

$$t_p = -\frac{K_c M_G \sin\theta_s + \sin(\theta_G + \theta_s)}{M_s \sin\theta_G} \tag{6-37}$$

即串联校正器$G_c(s) = K_c \dfrac{t_z s + 1}{t_p s + 1}$，便能够唯一确定。

【例 6-8】已知单位负反馈系统被控对象的传递函数为：

$$G_0(s) = \frac{256}{s(s+8)(s+16)}$$

试用根轨迹解析方法对系统进行超前串联校正设计，使之满足：

（1）阶跃响应的超调量$\sigma\% \leqslant 30\%$；

（2）阶跃响应的调节时间 $t_s \leqslant 0.8$ s；

（3）单位斜坡响应稳态误差$e_{ssv} \leqslant$ 10%。

【解】

（1）求校正器增益K_c

设校正后系统的传递函数为：

$$G_0(s)G_c(s) = \frac{256}{s(s+8)(s+16)} \cdot K_c \cdot \frac{t_z s + 1}{t_p s + 1} = K_c \cdot \frac{256}{s(s+8)(s+16)} \cdot \frac{t_z s + 1}{t_p s + 1}$$

根据 $e_{ssv} = \dfrac{v_0}{K_v} = \dfrac{1}{K_v} \leqslant$ 10%，故有 $K_v \geqslant 10$。

又根据自动控制的理论，有下式成立

$$K_v = \lim_{s\to 0} s \cdot G_0(s)G_c(s) = \lim_{s\to 0} s \cdot K_c \cdot \frac{256}{s(s+8)(s+16)} \cdot \frac{t_z s + 1}{t_p s + 1} \geqslant 10$$

有$K_c \geqslant 5$，取 $K_c = 5$。

（2）校检原系统的阶跃响应超调量是否满足要求

用以下程序 L6302a.m 绘制系统单位阶跃给定响应曲线并观察超调量。

```
% MATLAB PROGRAM L6302a.m
%
n1=1280;d1=conv(conv([1 0],[1 8]),[1 16]);
sope=tf(n1,d1);sys=feedback(sope,1);
step(sys)
```

运行程序后有图 6-24 所示的系统单位阶跃给定响应曲线。由曲线可知，系统阶跃响应超调近 50%，没有达到题目要求，所以需要进行校正。

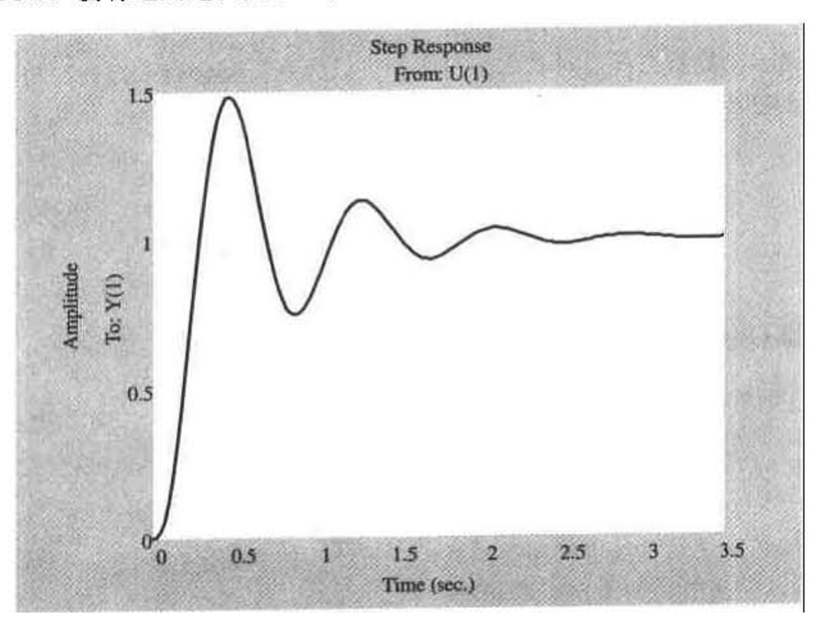


图 6-24　原系统阶跃给定响应曲线

（3）确定希望极点位置

根据题意，超调量$\sigma\% \leqslant 30\%$，求ς 。可输入以下 MATLAB 语句：

```
sigma=0.3;
zeta=((log(1/sigma))^2/((pi)^2+(log(1/sigma))^2))^(1/2)
```

运行后有

```
zeta = 0.3579
```

取$\varsigma = 0.4$，由$t_s(5\%) = \dfrac{3}{\varsigma\omega_n} = 0.8$，可得$\omega_n = 9.4$　rad/s，取$\omega_n = 10$　rad/s。可输入以下 MATLAB 语句求系统主导极点：

```
zeta=0.4;wn=10;
p=[1 2*zeta*wn wn*wn];
roots(p)
```

语句执行后得：

```
ans =
  -4.0000 + 9.1652i
  -4.0000 - 9.1652i
```

即主导极点：

$s_{1,2} = -4.0000 \pm 9.1652\,\mathrm{i}$

（4）求校正补偿器的传递函数

由式（6-32），校正器传递函数为：

$$G_c(s)=K_c\frac{t_z s+1}{t_p s+1}$$

根据根轨迹解析法校正理论，用以下程序 L6302b.m 求校正补偿器的传递函数。程序可改为用根轨迹解析法求超前校正器传递函数的函数 rtloc( )，使用这个函数需要已知未校正系统的开环传递函数与希望极点在 S 复平面上的位置以及校正器的增益 K_c，请参见本节后附录。

```
% MATLAB OROGRAM L6302b.m
%
kc=5;
s_1=-4.0000 +9.1652i;
nk1=256;
dk1=conv(conv([1 0],[1 8]),[1 16]);
ngv=polyval(nk1,s_1);
dgv=polyval(dk1,s_1);
g=ngv/dgv;
zetag=angle(g);zetag_d=zetag*180/pi;
mg=abs(g);ms=abs(s_1);
zetas=angle(s_1);zetas_d=zetas*180/pi;
tz=(sin(zetas)-kc*mg*sin(zetag-zetas))/(kc*mg*ms*sin(zetag));
tp=-(kc*mg*sin(zetas)+sin(zetag+zetas))/(ms*sin(zetag));
nk=[tz,1];dk=[tp,1];
Gc=tf(nk,dk)
```

运行程序后，可得校正器传递函数：

```
Transfer function:
0.09813 s + 1
-------------
 0.032 s + 1
```

运行以下调用 rtloc( )函数的程序可以得到相同的结果。

```
s1= -4.0000 + 9.1652i;kc=5;
n1=256;d1=conv(conv([1 0],[1 8]),[1 16]);
sope=tf(n1,d1);
[Gc]=rtloc(2,sope,[s1,kc]);
```

即有校正器传递函数：

$$G_c(s)=\frac{t_z s+1}{t_p s+1}=\frac{0.09813s+1}{0.032s+1}$$

（5）校检校正器计算是否正确

根据校正后系统的结构与参数，给出调用函数 perf( )的程序 L6302c.m 来实现计算。

```
% MATLAB PROGRAM L6302c.m
```

```
%
global y t nc dc
n1=1280;d1=conv(conv([1 0],[1 8]),[1 16]);
s1=tf(n1,d1);Gc=tf([0.09813 1],[0.032 1]);
sys=feedback(s1*Gc,1);step(sys);
[y,t]=step(sys);
[sigma,tp,ts]=perf(1,y,t);
```

运行该程序，可得阶跃响应曲线，如图 6-25 所示，并有

sigma = 26.3595 %

tp = 0.3589 s

ts = 0.5383 s

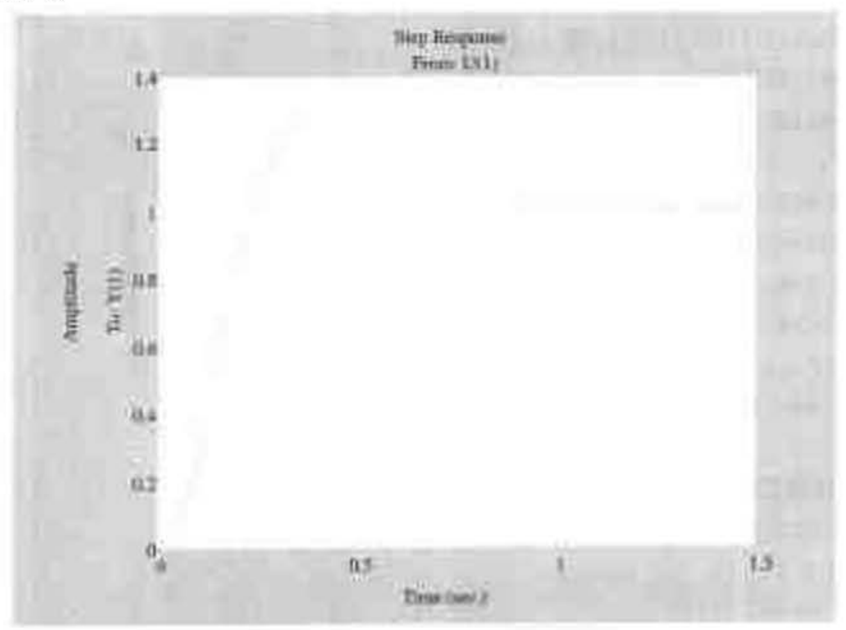


图 6-25　校正后系统的阶跃响应曲线

即系统经校正后，其阶跃响应品质指标是：超调量 $\sigma\% = 26.3595\% < 30\%$ 已达到要求，调节时间 $ts = 0.5383\ \mathrm{s} < 0.8\ \mathrm{s}$ 也已达到要求。即品质指标均合格。

6.3.2　根轨迹滞后校正设计

根轨迹滞后校正设计是为了使闭环主导极点在根轨迹上并满足静态指标要求，可以在根轨迹上添加一对偶极子，使其极点在其零点的右侧，从而使系统原根轨迹形状基本不变，而在期望主导极点处的稳态增益得到加大。下面先介绍根轨迹滞后校正的原理，然后介绍根轨迹滞后校正的步骤并举例。

6.3.2.1　根轨迹滞后校正设计的基本原理

设原系统的开环传递函数为：

$$G(s)H(s)=\frac{K_r\prod_{i=1}^{m}(s+z_i)}{s^{\upsilon}\prod_{j=\upsilon+1}^{n}(s+p_j)} \tag{6-38}$$

式中　z_i——系统前向通道传递函数的零点；

p_j——系统前向通道传递函数的极点；

K_r——开环系统根轨迹增益；

υ——系统型别。

根据系统根轨迹方程幅值条件，则根轨迹增益 K_r 为：

$$K_r=\frac{|s|^{\upsilon}\prod_{j=\upsilon+1}^{n}|s+p_j|}{\prod_{i=1}^{m}|s+z_i|} \tag{6-39}$$

将式(6-38)的 s 零次方项系数换算为 1,并由开环增益的定义，得系统开环增益 K 为：

$$K=K_r\frac{\prod_{i=1}^{m}z_i}{\prod_{j=\upsilon+1}^{n}p_j} \tag{6-40}$$

设滞后校正器传递函数为：

$$G_c(s)=\frac{1+Ts}{1+\beta Ts}$$

则校正系统的开环传递函数为：

$$G_c(s)\,G(s)H(s)=\frac{K_{rc}\prod_{i=1}^{m}(s+z_i)}{s^{\upsilon}\prod_{j=\upsilon+1}^{n}(s+p_j)}\cdot\frac{1+Ts}{1+\beta Ts} \tag{6-41}$$

当零点 $-1/T$ 与极点 $-1/\beta T$ 相对于点 s_1 是一对偶极子时，根据幅值条件则有：

$$K_{rc}=\frac{|s_1|^{\upsilon}\prod_{j=\upsilon+1}^{n}|s_1+p_j|}{\prod_{i=1}^{m}|s_1+z_i|}\cdot\beta\frac{\left|s_1+\frac{1}{T}\right|}{\left|s_1+\frac{1}{\beta T}\right|}\approx\beta\cdot\frac{|s_1|^{\upsilon}\prod_{j=\upsilon+1}^{n}|s_1+p_j|}{\prod_{i=1}^{m}|s_1+z_i|} \tag{6-42}$$

式中 s_1 为期望闭环主导极点，将 s_1 代入式（6-39），由式（6-43）则有：

$$K_{rc}=\beta\cdot K_r \tag{6-43}$$

对应于式（6-40），有校正系统静态开环增益：

$$K_c=K_{rc}\frac{\prod_{i=1}^{m}z_i}{\prod_{j=\upsilon+1}^{n}p_j}=\beta\cdot K_r\frac{\prod_{i=1}^{m}z_i}{\prod_{j=\upsilon+1}^{n}p_j} \tag{6-44}$$

考虑式（6-40）与（6-44）可得：

$$K_c = \beta K \tag{6-45}$$

故校正系统的静态开环增益增大为原系统开环增益的 β 倍。

6.3.2.2 根轨迹滞后校正设计的步骤

由上述基本原理，根轨迹滞后校正设计可归纳如下步骤：

（1）绘制未校正系统的根轨迹，根据动态品质指标要求，在根轨迹上确定期望闭环主导极点 s_1 的位置。

（2）根据式（6-39）确定在点 s_1 的根轨迹增益 K_r，根据式（6-40）确定在点 s_1 未校正系统的开环增益 K。

（3）计算系统所要求的静态误差系数 K_0。

（4）为了满足静态品质指标，计算误差系数所需增加的倍数：

$$\beta = \frac{K_0}{K} \tag{6-46}$$

（5）选择滞后校正器的零点 $-1/T$ 与极点 $-1/\beta T$，使之满足 $\frac{z_i}{p_i} = \beta$，这就要求 $-1/T$ 与 $-1/\beta T$ 相对于点 s_1 是一对偶极子，并距离坐标原点越近越好。

（6）校验所有性能指标是否合乎要求。

【例 6-9】已知单位负反馈系统被控对象的传递函数为：

$$G_0(s) = K_0 \frac{2500}{s(s+25)}$$

试用根轨迹解析方法对系统进行串联滞后校正设计，使之满足：

（1）阶跃响应的超调量 $\sigma\% \leqslant 15\%$；

（2）阶跃响应的调节时间 $t_s \leqslant 0.3$ s；

（3）单位斜坡响应稳态误差 $e_{ssv} \leqslant 0.01$。

【解】

（1）求静态误差系数 K_0

根据自动控制理论，对于 I 型系统有 $e_{ssv} = \frac{v_0}{K_v} = \frac{1}{K_v} \leqslant 0.01$，故有 $K_v \geqslant 100\ \mathrm{s}^{-1}$。取系统速度误差系数 $K_v = 100\ \mathrm{s}^{-1}$。

对于 I 型系统又有

$$K_v = \lim_{s \to 0} s \cdot \frac{2500 \cdot K_0}{s(s+25)} = 100$$

则 $K_0 = 1$

（2）校检原系统的阶跃响应超调量是否满足要求

根据系统闭环传递函数绘制单位阶跃给定响应曲线，用以下程序 L6303a.m 实现。

```
% MATLAB PROGRAM L6303a.m
%
n1 2500;d1=conv([1 0],[1 25]);
s1=tf(n1,d1);sys-feedback(s1,1);
step(sys)
```

运行程序后，有图 6-26 所示的系统阶跃响应曲线。

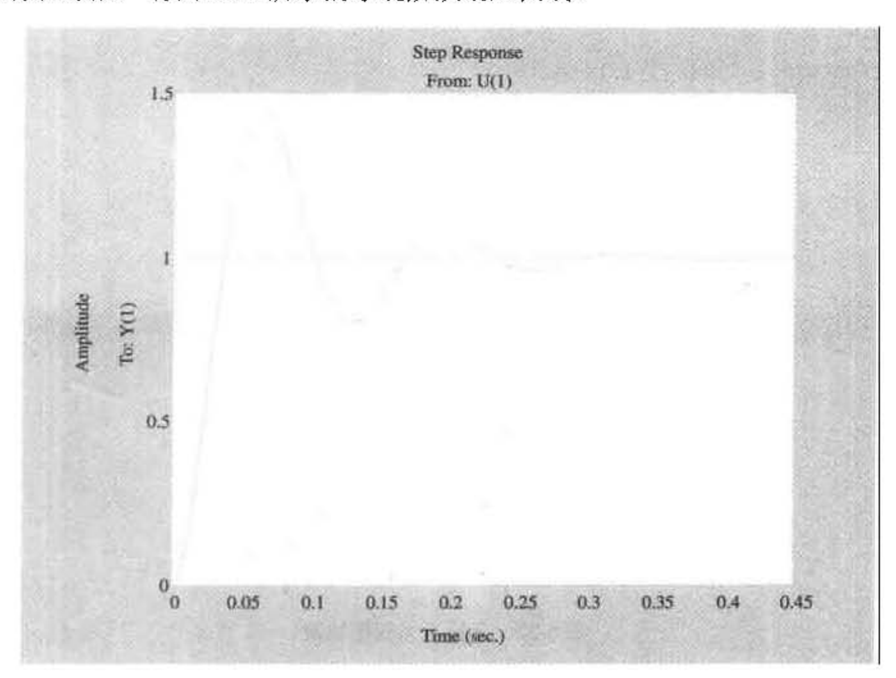


图 6-26　未校正系统的阶跃响应曲线

由阶跃响应曲线可知，系统阶跃响应超调量接近 50%，不满足题目超调量的要求。

（3）由期望极点位置确定校正器传递函数

为求校正器传递函数，需确定期望闭环主导极点 s_1 的位置，参看图 6-27。在图中直线 $s_1 0$ 与负实轴的夹角为 α，则有 $\varsigma=\cos\alpha$。用以下 MATLAB 语句求 ς。

```
sigma=0.15;
zeta=((log(1/sigma))^2/((pi)^2+(log(1/sigma))^2))^(1/2)
```

求得 zeta = 0.5169，取 zeta=0.54。

用以下 MATLAB 语句可绘制未校正系统的根轨迹图如图 6-27 所示。

```
n1=2500;d1=conv([1 0],[1 25])
s1=tf(n1,d1); rlocus(s1)
```

由图 6-27 可以看出，根轨迹分离点 d=－12.5。由 $G_0(s)$ 可知，未校正系统没有零点，有两个极点: p1=0,p2=－25。

用以下程序 L6303b.m 计算校正器传递函数。

```
% MATLAB PROGRAM L6303b.m
%
```

```
essv=0.01;x=-12.5;z1=0;p1=0;p2=25;
zeta=0.54;acos(zeta);ta=tan(acos(zeta));
y1=x*ta;y=abs(y1);s1=x+y*i;
Kr=abs(s1+p1)*abs(s1+p2);
K=Kr/(p1+p2);K0=1/essv;
beta=K0/K;T=1/((1/20)*abs(x));
betat=beta*T;
gc=tf((1/beta)*[1 1/T],[1 1/betat])
```

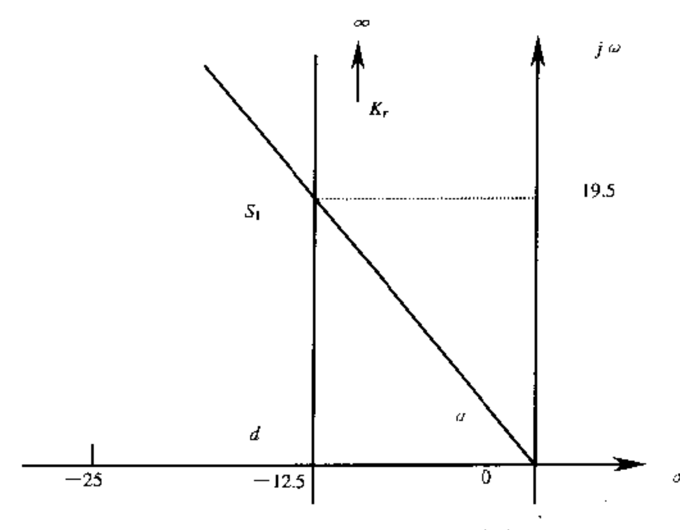


图 6-27　未校正系统根轨迹

程序运行后可得校正器传递函数为：

```
Transfer function:
0.2143 s + 0.134
----------------
  s + 0.134
```

校正系统传递函数为：

$$G_0(s)G_c(s)=\frac{2500}{s(s+25)}\cdot\frac{1}{\beta}\cdot\frac{s+\frac{1}{T}}{s+\frac{1}{\beta T}}=\frac{2500}{s(s+25)}\ \frac{0.2143s+0.134}{s+0.134}$$

（4）校检校正器计算是否正确

根据校正后系统的结构与参数，用以下程序 L6303c.m 来计算时域性能指标，程序中调用了函数 perf( )。

```
% MATLAB PROGRAM L6303c.m
%
s1=tf(2500,conv([1 0],[1 25]));
s2=tf([0.2143 0.134],[1 0.134]);
sope=s1*s2;sys=feedback(sope,1);
```

```
[y,t]=step(sys);[sigmaa,tp,ts]=perf(1,y,t);
```

程序运行后绘制的系统的阶跃响应曲线如图6-28所示，并计算出时域性能指标：

超调量：*sigma* = 15.5315 %

峰值时间：*tp* = 0.1581 s

调节时间：*ts* = 0.2258 s

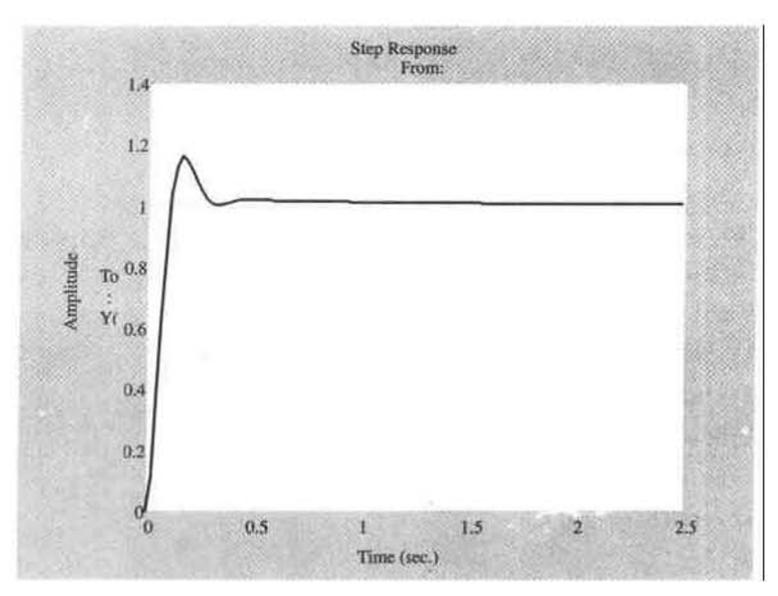


图6-28　校正后系统的阶跃响应曲线

由以上数据可知，超调量$\sigma\%$ =15.5313 %≈15 %已达到要求。调节时间ts= 0.2258s <0.3s也已达到要求。即品质指标均已基本达到题目要求。

附录：

【MATLAB FUNCTION PROGRAM 7】求系统串联超前校正器传递函数的函数rtloc( )

rtloc( )函数的调用格式为：

[*Gc*]=**rtloc**(*key*,sope,*vars*)

其中 sope 是系统的开环传递函数，*s*1是期望闭环主导极点，*kc*为校正器的增益。当*key*=1时，为用根轨迹几何方法求校正器，调用该函数时,*vars*为*s*1；当*key*=2时，为用根轨解析方法求校正器,调用该函数时,*vars*(1)为s1；*vars*(2)为*kc*；*s*1与*kc*都应放在方括号里,用逗号分隔。函数调用格式成为：

[*Gc*]=**rtloc(1,sope,[*s*1])**

[*Gc*]=**rtloc(2,sope,[*s*1,*kc*])**

```
function [Gc]=rtloc1(key,sope,vars)
% MATLAB FUNCTION PRAGRAM rtloc.m
%
num=sope.num{1};den=sope.den{1};s1=vars(1);
ngv=polyval(num,s1);
```

```
dgv=polyval(den,s1);
g=ngv/dgv;
if key==1
   zeta=angle(g);
    if zeta>0;
         phic=pi-zeta;
    end;
    if zeta<0;
         phic=-zeta;
    end;
   phi=angle(s1);
   zetaz=(phi+phic)/2;
   zetap=(phi-phic)/2;
   zc=real(s1)-imag(s1)/tan(zetaz);
   pc=real(s1)-imag(s1)/tan(zetap);
   nc=[1 -zc];dc=[1 -pc];
   nv=polyval(nc,s1);dv=polyval(dc,s1);
   kv=nv/dv;kc=abs(1/(g*kv));
    if zeta<0;
         kc=-kc;
    end;
   kc
   Gc=tf(nc,dc)
elseif key==2
   kc=vars(2);
   zetag=angle(g);
   mg=abs(g);ms=abs(s1);
   zetas=angle(s1);
   tz=(sin(zetas)-kc*mg*sin(zetag-zetas))/(kc*mg*ms*sin(zetag));
   tp=-(kc*mg*sin(zetas)+sin(zetag+zetas))/(ms*sin(zetag));
   nc=[tz,1];dc=[tp,1];Gc=tf(nc,dc)
end
```

6.4 控制系统 PID 校正器设计法

6.4.1 PID 调节简述

PID 调节又叫 PID 控制，是比例（Proportional）、积分(Integral)、微分(Differential)

调节的简称。在自动控制的发展历程中，PID 调节是历史最悠久、控制性能最强的基本调节方式。PID 调节原理简单，易于整定，使用方便；按照 PID 调节功能工作的各类调节器广泛应用于国民经济所有工业生产部门，适用性特强；PID 的调节性能指标对于受控对象特性的稍许变化不很敏感，这就极大地保证了调节的有效性；PID 调节可用于补偿系统使之达到大多数品质指标的要求。直到目前为止，PID 调节仍然是最广泛应用的基本控制方式。

具有 PID 特点的调节器既可以作为控制器，叫 PID 控制器；也可以作为校正器，叫 PID 校正器。它们都能发挥其独特的多项优越调节功能。

PID 校正是一种负反馈闭环控制。PID 校正器通常与被控对象串联连接，设置在负反馈闭环控制的前向通道上。

若校正器与系统的前向通道或者前向通道的一部分构成负反馈闭环连接，这种校正叫做反馈校正。

6.4.2 PID 调节规律

在图 6-29 中，给出了作为校正器的 PID 调节动态结构框图。在 PID 调节作用下，对误差信号 $e(t)$ 分别进行比例、积分、微分运算，三个作用分量之和作为控制信号输出给被控制对象。图中信号为其对应量的拉氏变换。

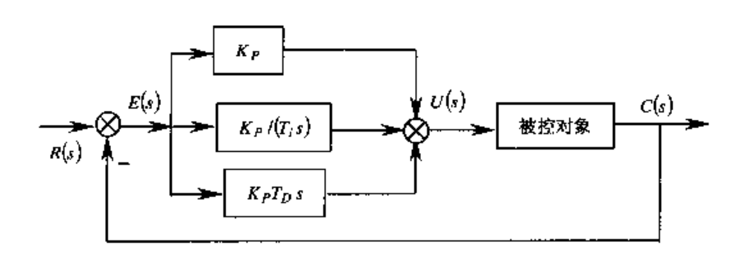


图 6-29　典型 PID 校正器结构框图

PID 调节器的微分方程数学模型为：

$$u(t)=K_P\left[e(t)+\frac{1}{T_I}\int_0^t e(t)+T_D\frac{\mathrm{d}e(t)}{\mathrm{d}t}\right] \tag{6-47}$$

式中 $u(t)$ 为 PID 调节器的输出信号，这个信号就是用来送给受控对象的；系统误差信号定义为：$e(t)\overset{\text{def}}{=}r(t)-c(t)$，$r(t)$ 是系统的给定输入信号；$c(t)$ 是系统的被控量。

PID 调节的传递函数模型

$$G_c(s)=K_P\left(1+\frac{1}{T_I s}+T_D s\right) \tag{6-48}$$

由式（6-48）可得 PID 调节的几种特例形式：

对于式（6-48），当 $T_D=0$、$T_I=\infty$ 时，则有 $G_c(s)=K_P$，此为比例（P）调节器；当 $T_I=\infty$ 时，则有 $G_c(s)=K_P(1+T_D s)$，此为比例微分（PD）调节器，若将其作为校正器，

它相当于超前校正器；而当$T_D=0$时，则有$G_c(s)=K_P\left(1+\dfrac{1}{T_I s}\right)$，此为比例积分（PI）调节器，若将其作为校正器，它相当于滞后校正器；当$K_P\neq 0$、$T_D\neq 0$、$T_I\neq\infty$时，则有$G_c(s)=K_P(1+\dfrac{1}{T_I s}+K_D s)$，这叫做全PID调节器。

6.4.3 PID调节作用分析

由式（6-48）可以看出，PID控制是通过三个参量K_P、T_I、T_D起作用的。这三个参量取值的大小不同，就是比例、积分、微分作用强弱的变化。为了说明每个参量单独变化时对于系统校正作用的影响，特举以下示例。以下在讨论某一个变量变化产生的影响时，将保持另外两个参量为某一常数不变，这样得出的结论，概念更加清晰。

【例6-10】续【例5-1】将晶闸管直流单闭环调速系统的转速调节器改换为PID调节器，如图6-30所示。试对调速系统的P、I、D控制作用进行分析。

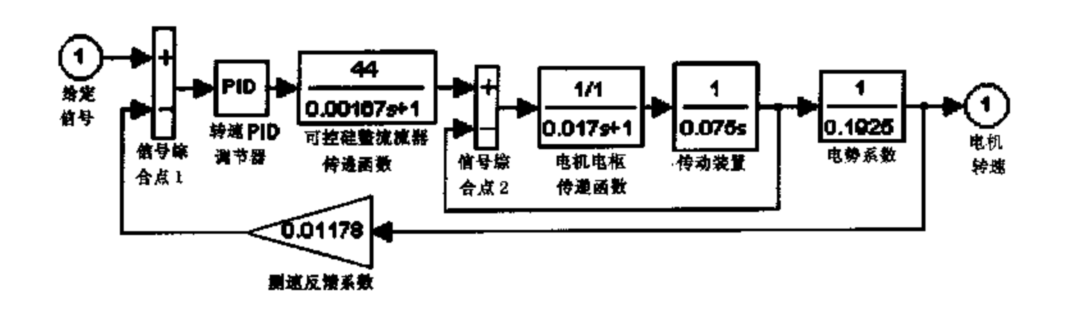


图6-30 单闭环调速系统

【解】

（1）比例调节作用分析

为分析纯比例调节的作用，考查当$T_D=0$、$T_I=\infty$、Kp=1~5时对系统阶跃给定响应的影响，根据图6-30 结构图的数据，给出以下MATLAB程序L6401a.m。

```
% MATLAB PROGRAM L6401a.m
%
G1=tf(1,[0.017 1]);
G2=tf(1,[ 0.075 0]);
G12=feedback(G1*G2,1);
G3=tf(44,[0.00167 1]);
G4=tf(1,0.1925);
G=G12*G3*G4;
Kp=[1:1:5];
for i=1:length(Kp)
  Gc=feedback(Kp(i)*G,0.01178);
```

```
    step(Gc),hold on
end
axis([0,0.2,0,130]);
gtext('1 Kp=1'),
gtext('2 Kp=2'),
gtext('3 Kp=3'),
gtext('4 Kp=4'),
gtext('5 Kp=5'),
```

运行程序后有系统比例(P)调节器作用下系统阶跃给定响应曲线，如图 6-31 所示。

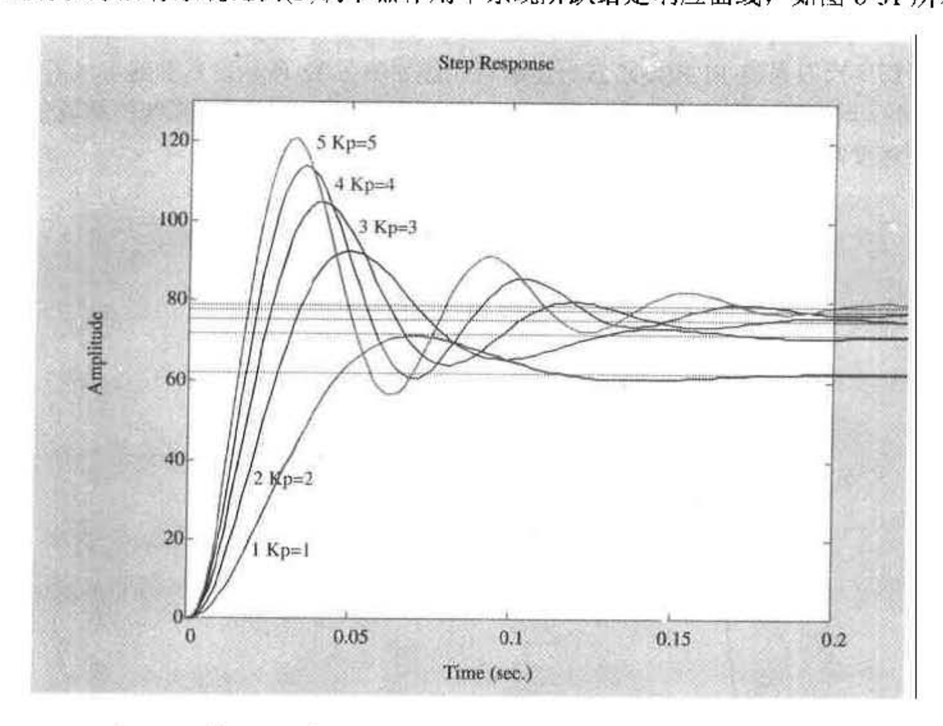


图 6-31　单闭环调速系统 P 控制阶跃给定响应曲线

从系统 P 调节阶跃给定响应曲线图 6-31 可以看出，随着 K_p 值的加大，闭环系统的超调量加大，系统响应速度加快。仿真还表明，当 $K_p \geq 21$ 后，系统变为不稳定。

（2）积分调节作用分析

为分析方便起见，对比例积分调节器保持 $K_p=1$ 时，考查当 $T_I=0.03\sim0.07$ 时对系统阶跃给定响应的影响，根据 图 6-30 的数据，给出 MATLAB 程序 L6401b.m 如下。

```
% MATLAB PROGRAM L6401b.m
%
G1=tf(1,[0.017 1]);G2=tf(1,[ 0.075 0]);
G12=feedback(G1*G2,1);G3=tf(44,[0.00167 1]);
G4=tf(1,0.1925);G=G12*G3*G4;
kp=1;Ti=[0.03:0.01:0.07];
```

```
for i=1:length(Ti)
   Gc=tf(kp*[Ti(i) 1],[Ti(i) 0]);
   Gcc=feedback(G*Gc,0.01178);
   step(Gcc),hold on
end
gtext('1 Ti=0.01'),
gtext('2 Ti=0.02'),
gtext('3 Ti=0.03'),
gtext('4 Ti=0.04'),
gtext('5 Ti=0.05'),
```

运行程序后有系统 PI 控制阶跃给定响应曲线如图 6-32 所示。从曲线可以看出，保持 K_p=1 不变时，在本程序设定值的范围内，随着 T_I 值的加大，闭环系统的超调量减小，系统响应速度略微变慢。

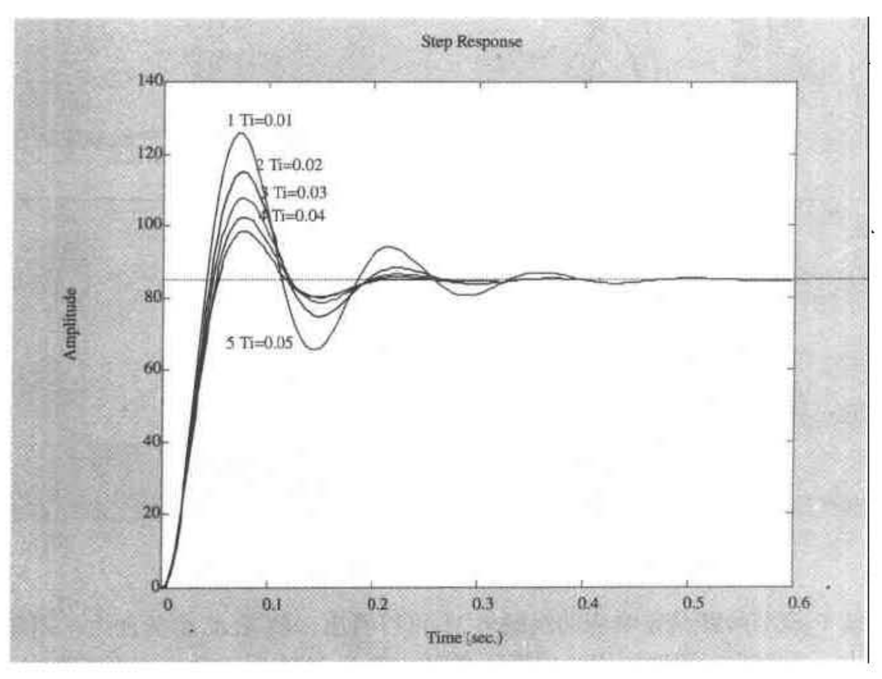


图 6-32　单闭环调速系统 PI 控制阶跃给定响应曲线

（3）微分调节作用分析

为分析方便起见，对比例积分微分调节器保持 K_p=0.01、$T_I=0.01$ 时，特别考查当 $T_D=12\sim84$ 时对系统阶跃给定响应的影响，由结构图 6-32 的数据，给出 MATLAB 程序 L6401c.m 如下：

```
% MATLAB PROGRAM L6401c.m
%
G1=tf(1,[0.017 1]);G2=tf(1,[ 0.075 0]);
G12=feedback(G1*G2,1);G3=tf(44,[0.00167 1]);
```

```
G4=tf(1,0.1925);G=G12*G3*G4;
Kp=0.01;Ti=0.01;
Td=[12:36:84];
for i=1:length(Td)
  Gc=tf(Kp*[Ti*Td(i) Ti 1],[Ti 0]);
  Gcc=feedback(G*Gc,0.01178);
  step(Gcc),hold on
end
gtext('1 Td=12'),
gtext('2 Td=48'),
gtext('3 Td=84'),
```

运行程序后可得系统 PID 控制阶跃给定响应曲线，如图 6-33 所示。从图 6-33 可看出：第一，由于单闭环调速系统的参数配合的特殊性及微分环节的作用，在曲线的起始上升段呈现尖锐的波峰，之后曲线也呈衰减的振荡；第二，当保持 K_P=0.01、T_I =0.01 不变时，在本程序设定的T_D范围内(T_D = 12:36:84)，随着T_D值的加大，闭环系统的超调量增大，但经曲线尖锐的起始上升段后响应速度有所变慢。

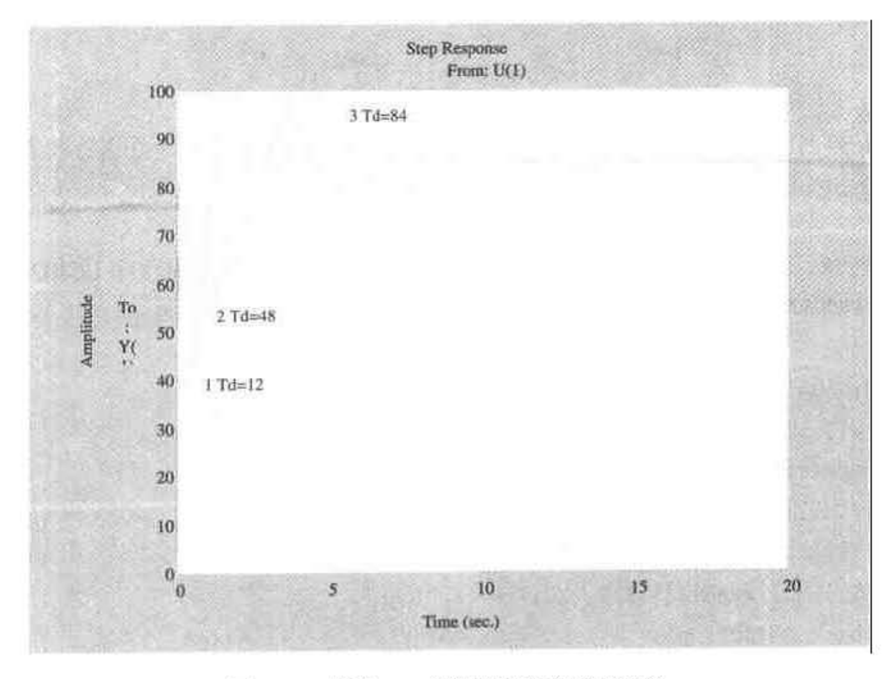


图 6-33 系统 PID 控制阶跃给定响应曲线

6.4.4 PID 校正器设计方法

传统 PID 控制的经验公式是 Ziegler 与 Nichols 在 20 世纪 40 年代初提出的，有一定的实用价值。Ziegler — Nichols 公式临界振荡原则的算法； PID 控制的各种算法及程序[17]，这些算法都在不同要求的领域里得到了有效的应用。

6.4.4.1 动态特性参数法（Ziegler –Nichols 整定公式）的 PID 校正器设计

对于被控广义对象为带延迟的一阶惯性环节的系统，即传递函数表达式为式（5-11）的系统，其 PID 控制的参数值可以用一组经验公式来计算。这种 PID 调节器参数值确定的方法是 1942 年由齐格勒（Ziegler）与尼柯尔斯（Nichols）首先提出的。已知被控对象的传递函数模型 $G(s)=\dfrac{K}{Ts+1}e^{-\tau s}$ 的三个参数 K、T、τ 时，整定 PID 调节器参数的计算公式如表 6-1 所示。

表 6-1　调节器 Ziegler –Nichols 整定公式

	K_P	T_I	T_D
P	$T/(K\tau)$		
PI	$0.9T/(K\tau)$	3.3τ	
PID	$1.2T/(K\tau)$	2.2τ	0.5τ

为实现用 Ziegler –Nichols 整定公式计算系统 P、PI、PID 校正器的参数，作者给出函数 zn01.m。函数 zn01( )参见本节后“补充”。

【例 6-11】已知过程控制系统的被控广义对象为一个带延迟的惯性环节，其传递函数为：

$$G_0(s)=\frac{8}{360s+1}e^{-180s}$$

试用 Ziegler –Nichols 整定公式计算系统 P、PI、PID 校正器的参数，并进行阶跃给定响应的仿真。

【解】

根据题意，利用 zn01( ) 函数计算系统 P、PI、PID 校正器参数的程序 L6402.m 如下。

```
% MATLAB PROGRAM 6402.m
%
K=8;T=360;tau=180;
n1=[K];d1=[T 1];G1=tf(n1,d1);
[np,dp]=pade(tau,2);Gp=tf(np,dp);
[Gc1,Kp1]=zn01(1,[K,T,tau])
[Gc2,Kp2,Ti2]=zn01(2,[K,T,tau])
[Gc3,Kp3,Ti3,Td3]=zn01(3,[K,T,tau])
Gcc1=feedback(G1*Gc1,Gp);
set(Gcc1,'Td',tau);step(Gcc1);hold on
Gcc2=feedback(G1*Gc2,Gp);
set(Gcc2,'Td',tau);step(Gcc2);
Gcc3=feedback(G1*Gc3,Gp);
set(Gcc3,'Td',tau);step(Gcc3);
gtext('1 P control'),gtext('2 PI control'),
gtext('3 PID control')
```

程序运行后，求出 P、PI、PID 校正器的传递函数分别为：

```
Gc1 = 0.2500
Gc2 =
134.9 s + 0.225
---------------
   599.4 s
Gc3 =
9720 s^2 + 108 s + 0.3
----------------------
 360 s
```

程序运行后，还得到经 P、PI、PID 校正（用 Ziegler–Nichols 公式计算）系统阶跃给定响应曲线，如图 6-34 所示。

由图 6-34 可见，用 Ziegler –Nichols 公式计算的 P、PI、PID 校正器对系统校正后，其阶跃给定响应曲线中的 P、PI 校正两者响应速度基本相同。因为这两种校正求出的 *Kp* 不同，所以两种校正的被调量终了值不同。PI 校正的超调量比 P 校正的要小些。PID 校正的比前两者的响应速度都要快，但是超调量最大。

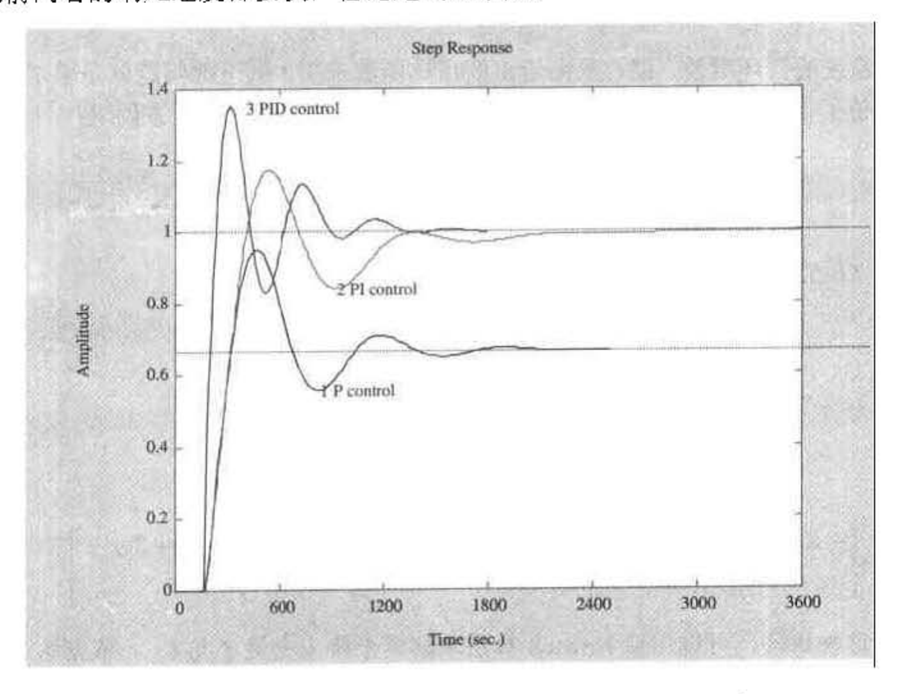


图 6-34　Ziegler –Nichols 的 P、PI、PID 控制阶跃响应曲线

6.4.4.2　一般数学模型拟合成带延迟的惯性环节

被控广义对象的传递函数模型为 $G(s)=\dfrac{K}{Ts+1}e^{-\tau s}$ 这种表达式形式，是使用经典 Ziegler –Nichols 整定公式设计 PID 校正器的前提。如果已知系统模型不是这种形式的，

可以将模型经过转换计算求其模型拟合的对应参数 K 、T 、τ 。许多文献中都介绍过这样的算法，不同的方法各有所长。基于频域响应的 Jacobian 矩阵求解方法，概念清楚，计算简单，使用方便，有较强的实用性[17]。其理论基础仍然是模型的频率特性。

对于传递函数为 $G(s)=\dfrac{K}{Ts+1}e^{-\tau s}$ 的数学模型，其频率特性为：

$$G(j\omega)=G(s)\big|_{s=j\omega}=\frac{K}{Ts+1}e^{-\tau s}\bigg|_{s=j\omega}=\frac{K}{Tj\omega+1}e^{-j\omega\tau} \tag{6-49}$$

考虑尤拉公式

$$e^{-j\omega\tau}=[\cos(\omega\tau)-j\sin(\omega\tau)] \tag{6-50}$$

当系统乃奎斯特（开环幅相特性）曲线在其复平面与负实轴相交时，其第一个交点模值的倒数为系统模值稳定裕量 Gm （非分贝值），其交点对应的角频率为-π穿越频率 ω_{cg} 。因为交点是在负实轴上，交点对应复变量的虚部为零，即有以下方程组：

$$\begin{cases}\dfrac{K[\cos(\omega_{cg}\tau)-\omega_{cg}T\sin(\omega_{cg}\tau)]}{1+\omega_{cg}^2T^2}=-\dfrac{1}{Gm}\\ \sin(\omega_{cg}\tau)+\omega_{cg}T\cos(\omega_{cg}\tau)=0\end{cases} \tag{6-51}$$

K 是系统的开环增益，很容易由给出的传递函数求出。剩下的问题就是求 T 与 τ 两个变量的值了。将式（6-51）改写为以 T 与 τ 为待求未知量的以下方程组：

$$\begin{cases}f_1(\tau,T)=KK_c[\cos(\omega_{cg}\tau)-\omega_{cg}T\sin(\omega_{cg}\tau)]+1+\omega_{cg}^2T^2=0\\ f_2(\tau,T)=\sin(\omega_{cg}\tau)+\omega_{cg}T\cos(\omega_{cg}\tau)=0\end{cases} \tag{6-52}$$

由方程组（6-52），有其对应的 Jacob 矩阵：

$$J=\begin{bmatrix}\dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2}\\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2}\end{bmatrix}$$

$$=\begin{bmatrix}-KK_c\omega_{cg}\sin(\omega_{cg}\tau)-KK_c\omega_{cg}^2T\cos(\omega_{cg}\tau) & -KK_c\omega_{cg}\sin(\omega_{cg}\tau)+2\omega_{cg}^2T\\ \omega_{cg}\cos(\omega_{cg}\tau)-\omega_{cg}^2T\sin(\omega_{cg}\tau) & \omega_{cg}\cos(\omega_{cg}\tau)\end{bmatrix} \tag{6-53}$$

对于这种矩阵，可以用拟 Newton 算法求解两个待求变量 T 与 τ 。依据这个原理，作者给出函数 kttau( )用来求解 K、T 与 τ 。函数 kttau( )参见本节附录。

6.4.4.3 Cohen-Coon 整定公式的 PID 校正器设计

传统的 Ziegler –Nichols 整定公式经过改进，出现了各种不同设计 PID 控制器的算法，其中 Cohen-Coon 整定公式与传统的 Ziegler –Nichols 整定公式很类似。只要知道系统被拟合成带延迟惯性环节的参数 K、T 与 τ （用函数 kttau( )解决），则可以由表 6-2 中给出计算公式求得的数据直接设计出 PID 校正器。

表 6-2　调节器 Cohen-Coon 整定公式

	K_P	T_I	T_D
P	$\frac{1}{K}\left[(\tau/T)^{-1}+0.333\right]$		
PI	$\frac{1}{K}\left[0.9(\tau/T)^{-1}+0.082\right]$	$T\frac{3.33(\tau/T)+0.3(\tau/T)^2}{1+2.2(\tau/T)}$	
PD	$\frac{1}{K}\left[1.24(\tau/T)^{-1}+0.1612\right]$		$T\frac{0.27(\tau/T)}{1+0.13(\tau/T)}$
PID	$\frac{1}{K}\left[1.35(\tau/T)^{-1}+0.27\right]$	$T\frac{2.5(\tau/T)+0.5(\tau/T)^2}{1+0.6(\tau/T)}$	$T\frac{0.37(\tau/T)}{1+0.2(\tau/T)}$

作者给出函数 cc01( )来实现用 Cohen-Coon 整定公式计算系统 P、PI、PID 校正器的参数，函数 cc01( )参见本节后“附录”。

【例 6-12】某温度过程控制系统被控广义对象的传递函数为：

$$G_0(s)=\frac{K}{(Ts+1)^2}e^{-\tau s}$$

式中　K=1° C/kg * min^{-1}，　T=30 s，τ =60π/4 s ，试用 Cohen-Coon 整定公式计算系统串联 P、PI、PD、PID 校正器的参数，并进行阶跃给定响应的仿真。

【解】

利用函数 kttau( )先求被控对象传递函数拟合成带延迟-惯性环节的三个参数 K 、T 、τ 。根据题意，利用 cc01( ) 函数计算系统 P、PI、PD、PID 校正器参数的程序 L6403.m 如下。

```
% MATLAB PROGRAM L6403.m
%
G1=tf(1/60,[30 1]);G2=tf(1,[30 1]);tau1=60*pi/4;
[np,dp]=pade(tau1,2);Gp=tf(np,dp);G=G1*G2*Gp;
[K,T,tau]=kttau(G);
[Gc1]=cc01(1,[K,T,tau])
[Gc2]=cc01(2,[K,T,tau])
[Gc3]=cc01(3,[K,T,tau])
[Gc4]=cc01(4,[K,T,tau])
G12=G1*G2;
Gcc1=feedback(G12*Gc1,Gp);
set(Gcc1,'Td',tau1);step(Gcc1);hold on
Gcc2=feedback(G12*Gc2,Gp);
set(Gcc2,'Td',tau1);step(Gcc2);
Gcc3=feedback(G12*Gc3,Gp);
set(Gcc3,'Td',tau1);step(Gcc3);
Gcc4=feedback(G12*Gc4,Gp);
set(Gcc4,'Td',tau1);step(Gcc4);
axis([0,1000,0,1.5]);
gtext('1 P control'),
```

```
gtext('2 PI control'),
gtext('3 PD control'),
gtext('4 PID control')
```

程序运行后，求出 P、PI、PD 、PID 校正器的传递函数分别为：

```
Gc1 = 69.9614
Gc2 =
3160 s + 49.9
-------------
  63.31 s
Gc3 =
2395 s + 71.65
Gc4=
3.208e005 s^2 + 9420 s + 83.67
------------------------------
    112.6 s
```

程序运行后，还得到用 Cohen-Coon 公式计算的 P、PI、PD 、PID 校正器校正后系统阶跃给定响应曲线，如图 6-35 所示。

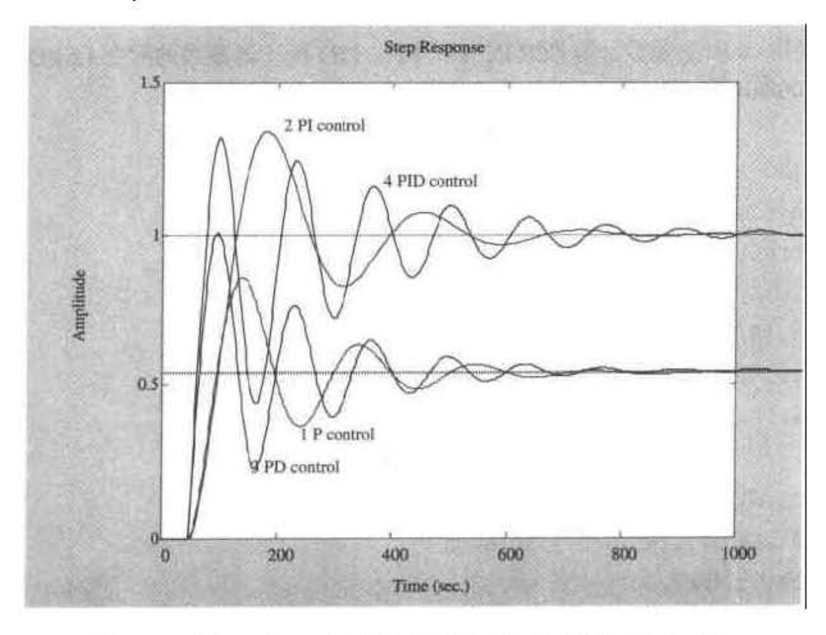


图 6-35　Cohen-Coon 公式计算的三种校正阶跃给定响应曲线

由图 6-35 可见，用 Cohen-Coon 公式计算的 P、PI、PD、PID 校正器对系统校正后，其阶跃给定响应曲线中的 P、PI 校正两者响应速度基本相同，两种校正的超调量大不一样。

而 PD、PID 两种校正的响应速度基本相同，两种校正的超调量也大不一样。因为 P、PD 与 PI、PID 两类情况的 *Kp* 不同，所以被调量的终了值不同。PD 与 PID 校正的响应速度比另外两种快，P 与 PD 两种校正的超调量比另外两种大。

6.4.4.4 最优控制（误差积分指标最优）的 PID 校正器设计

根据已知或者拟合的控制过程特性参数 K、T、τ，用误差最优积分的三个特例形式，即式（5-5）的 IAE、式（5-6）的 ISE、式（5-7）的 ITAE，以其积分值最小为准则，可以计算 PID 校正器整定参数的最优值。计算 PID 校正器传递函数仍为式（6-48）：

$$G_c(s)=K_P\left(1+\frac{1}{T_I s}+T_D s\right)$$

对于恒值控制系统，调节器参数计算公式为

$$Y_n=A(\tau/T)^B \tag{6-54}$$

对于不同的控制作用 Y_n 有不同的形式，即

比例调节作用 $Y_1=K_pK$ （6-55）

积分调节作用 $Y_2=T/T_I$ （6-56）

微分调节作用 $Y_3=T_D/T$ （6-57）

参变量 A、B 的数值列于表 6-3。

表 6-3 恒值系统按最小积分整定的 A、B 值

	控制规律	控制作用	A	B
IAE	P	P	0.902	-0.985
ISE	P	P	1.414	-0.917
ITAE	P	P	0.904	-1.084
IAE	PI	P	0.984	-0.986
IAE	PI	I	0.608	-0.707
ISE	PI	P	1.305	-0.959
ISE	PI	I	0.492	-0.739
ITAE	PI	P	0.859	-0.977
ITAE	PI	I	0.674	-0.680
IAE	PID	P	1.435	-0.921
IAE	PID	I	0.878	-0.749
IAE	PID	D	0.482	1.137
ISE	PID	P	1.495	-0.945
ISE	PID	I	1.101	-0.771
ISE	PID	D	0.560	1.006
ITAE	PID	P	1.357	-0.947
ITAE	PID	I	0.842	-0.738
ITAE	PID	D	0.381	0.995

为了实现用积分值最小准则来计算系统 P、PI、PID 校正器的参数，作者给出函数 inmin01( )。函数 inmin01( )参见本节后“附录”。

【例 6-13】已知某被控广义对象的传递函数为：

$$G_0(s)=\frac{1}{20s+1}e^{-2.5s}$$

试用 IAE、ISE、ITAE 积分值最小准则计算系统串联 P、PI、PID 校正器的参数，并对其进行阶跃给定响应的仿真。

【解】

（1）计算比例(P)校正器并作阶跃响应仿真

根据题意，利用 inmin01( ) 函数计算系统 P 校正器参数的程序 L6404a.m 如下。

```
% MATLAB PROGRAM L6404a.m
%
K=0.5;T=8;tau=5;
[np,dp]=pade(tau,2);
Gp=tf(np,dp);
G=tf(0.5,[8 1]);
iC=1:3;
for i=1:length(iC)
   [Gc1,Kp]=inmin01(1,[K,T,tau,iC(i)])
   Gc=G*Gc1;Gcc=feedback(Gc,Gp);
   set(Gcc,'Td',tau);
   step(Gcc),hold on
   axis([0,50,0,1.4]);
end
gtext('IAE'),
gtext('ISE'),
gtext('ITAE'),
```

程序运行后，可以得到三种最优控制下实现 P 校正时的参数：

①IAE 积分最小准则校正器：$Gc1 = Kp = 2.8661$

②ISE 积分最小准则校正器：$Gc1 = Kp = 4.3517$

③ITAE 积分最小准则校正器：$Gc1 = Kp = 3.0093$

运行程序后还得到三种积分最小准则 P 校正的阶跃给定响应曲线如图 6-36 所示。三条曲线有三个不同的终了值，IAE 的最小，ITAE 次之，ISE 的最大。这可以从计算出的 P 校正器 Kp 的不同大小值看出。IAE 与 ITAE 这两种曲线接近重合，响应速度与超调都差不多。而 ISE 的响应速度最快，超调量也最大。

（2）计算 PI 校正器并作阶跃响应仿真

根据题意，利用 inmin01( ) 函数计算系统 PI 校正器参数的程序 L6404b.m 如下。

```
% MATLAB PROGRAM L6404b.m
%
K=0.5;T=8;tau=5;
[np,dp]=pade(tau,2);
Gp=tf(np,dp);
G=tf(0.5,[8 1]);
iC=1:3;
for i=1:length(iC)
    [Gc2,Kp,Ti]=inmin01(2,[K,T,tau,iC(i)])
   Gc=G*Gc2;
```

```
    Gcc=feedback(Gc,Gp);
    set(Gcc,'Td',tau);
    step(Gcc),hold on
    axis([0,100,0,1.6]);
end
gtext('IAE'),
gtext('ISE'),
gtext('ITAE'),
```

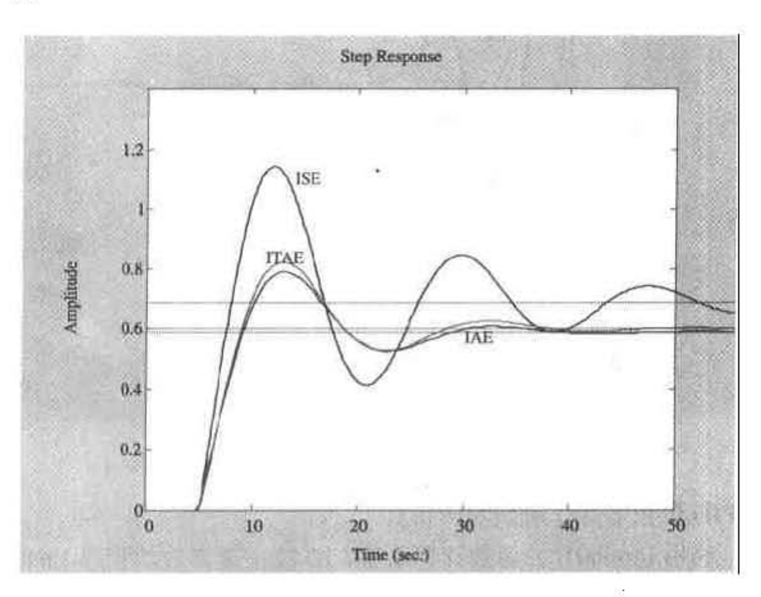


图 6-36 积分最小准则 P 校正阶跃给定响应曲线

程序运行后，可以得到三种最优控制下实现 **PI** 校正时的参数：

①IAE 积分最小准则校正器:$Kp = 3.1281$；$Ti = 9.4379$

```
Transfer function:
29.52 s + 3.128
---------------
   9.438 s
```

②ISE 积分最小准则校正器：$Kp = 4.0963$；$Ti = 11.4889$

```
Transfer function:
47.06 s + 4.096
---------------
   11.49 s
```

③ITAE 积分最小准则校正器：$Kp = 2.7192$；$Ti = 8.6224$

```
Transfer function:
23.45 s + 2.719
---------------
```

8.622 s

运行程序后还得到三种积分最小准则 PI 校正，如图 6-37 所示阶跃给定响应曲线。三条曲线的终了值是一致的。IAE 与 ITAE 这两种的曲线接近重合，响应速度与超调相差不多，IAE 的稍大一点。而 ISE 的响应速度较快，超调量也最大。

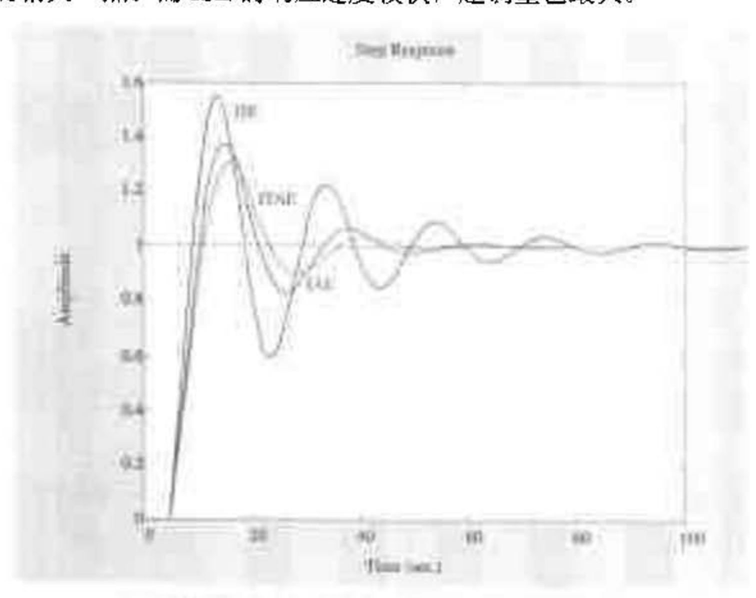


图6-37 积分最小准则PI校正阶跃给定响应曲线

（3）计算 PID 校正器并作阶跃响应仿真

根据题意，利用 inmin01( ) 函数计算系统 PID 校正器参数的程序 L6404c.m 如下。

```
% MATLAB PROGRAM L6404c.m
%
K=0.5;T=8;tau=5;
[np,dp]=pade(tau,2);
Gp=tf(np,dp);
G=tf(0.5,[8 1]);
iC=1:3;
for i=1:length(iC)
   [Gc3,Kp,Ti,Td]=inmin01(3,[K,T,tau,iC(i)])
   Gc=G*Gc3;Gcc=feedback(Gc,Gp);
   set(Gcc,'Td',tau);
   step(Gcc),hold on
   axis([0,50,0,2.0]);
end
gtext('IAE'),
gtext('ISE'),
gtext('ITAE'),
```

程序运行后，可以得到三种最优控制下实现 PID 校正时的参数：

①IAE 积分最小准则校正器：$Kp = 4.4246$；$Ti = 6.4078$；$Td = 2.2597$

```
Transfer function:
64.07 s^2 + 28.35 s + 4.425
---------------------------
   6.408 s
```

②ISE 积分最小准则校正器：$Kp = 4.6619$；$Ti = 5.0574$；$Td = 2.7921$

```
Transfer function:
65.83 s^2 + 23.58 s + 4.662
---------------------------
   5.057 s
```

③ITAE 积分最小准则校正器：$Kp = 4.2356$；$Ti = 6.7164$；$Td = 1.9095$

```
Transfer function:
54.32 s^2 + 28.45 s + 4.236
---------------------------
   6.716 s
```

运行程序后还得到三种积分最小准则 PID 校正，阶跃给定响应曲线如图 6-38 所示。三条曲线的终了值是一致的。就响应速度而言，以 ITAE、IAE、ISE 的顺序依次略为加快；超调相则以同样的顺序加大，ISE 的超调量最大。

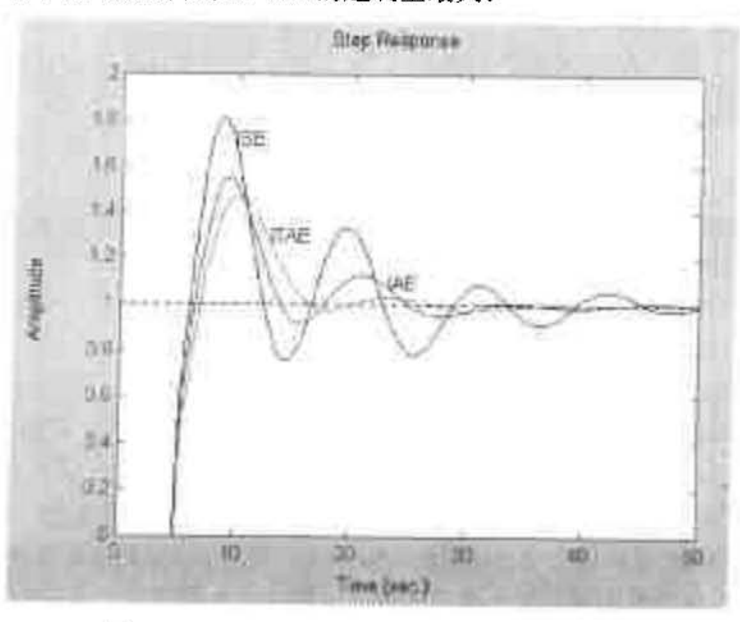


图 6-38　积分最小准则 PID 校正阶跃给定响应曲线

6.4.4.5　稳定边界法的 PID 校正器设计

稳定边界法，是目前应用较广的一种控制器参数计算方法。这个方法乃是基于系统的稳定性理论。系统闭环特征方程的根（即闭环极点）都在其复平面虚轴的左侧时，闭环系统稳

定；当闭环特征方程有纯虚根时，系统的根轨迹与虚轴相交，其响应等幅振荡，系统临界稳定。当置 PID 调节器的$T_I=\infty$与$T_D=0$时，增加K_P值直至系统开始振荡，此时系统闭环极点应在复平面的 $j\omega$ 虚轴上，确定系统闭环根轨迹与复平面 $j\omega$ 轴交点，求出交点的振荡角频率ω_m及其对应的系统增益K_m，则其 PID 调节器参数整定计算公式如表 6-4 所示。

表 6-4　稳定边界法 PID 整定公式

	K_P	T_I	T_D
P	$0.5K_m$		
PI	$0.455K_m$	$0.85*2\pi/\omega_m$	
PID	$0.6K_m$	$0.50*2\pi/\omega_m$	$0.125*2\pi/\omega_m$

为了实现用稳定边界法整定公式计算系统 P、PI、PID 校正器的参数，给出函数 zn02( )。函数 zn02( )参见本节后“附录”。

【例 6-14】已知一串级过程控制系统的主、副被控对象与副调节器的传递函数分别为：

$$G_{p1}(s)=\frac{1}{(30s+1)(3s+1)}；\ G_{p2}(s)=\frac{1}{(10s+1)(s+1)^2}；\ G_{c2}(s)=10$$

试用稳定边界法计算系统主调节器$G_{c1}(s)$作 P、PI、PID 校正时的参数，并进行阶跃给定响应的仿真。

【解】

根据题意，利用 zn02( ) 函数求系统 PI 校正器参数的程序 L6405.m 如下。

```
% MATLAB PROGRAM L6405.m
%
G1=tf(1,[30 1]);G2=tf(1,[3 1]);G3=tf(1,[10 1]);
G4=tf(10,[1 2 1]);G=G1*G2*G3*G4;
[Gc1,Kp1]=zn02(1,G)
[Gc2,Kp2,Ti2]=zn02(2,G)
[Gc3,Kp3,Ti3,Td3]=zn02(3,G)
Gcc1=feedback(G*Gc1,1);step(Gcc1);hold on
Gcc2=feedback(G*Gc2,1);step(Gcc2);
Gcc3=feedback(G*Gc3,1);step(Gcc3);
gtext('1 P control'),gtext('2 PI control'),
gtext('3 PID control')
```

程序须在 MATLAB 命令窗口中运行，程序运行后有根轨迹图 6-39，图上显示有十字光标，选择根轨迹与虚轴的交点用鼠标左键点击。再回到 MATLAB 命令窗口中，可以见到有计算出的根轨迹增益与极点值（应该力求准确定在根轨迹与虚轴的交点上，尽量使极点实部为 0），还可见到字符“K”。应该在字符“K”后键入指令“return”，然后即可在 MATLAB 命令窗口中看到 km = 1.1436, wm = 0.1646，这就是交点的系统增益K_m及其对应的振荡角频率ω_m。在 MATLAB 命令窗口中还看到用稳定边界法计算出 P、PI、PID 校正的参数。程序三次调用函数 zn02( )，需如此操作三次，最后得到如图 6-40 所示的 PID 三种校正时的阶跃给定响应曲线与校正器计算结果。

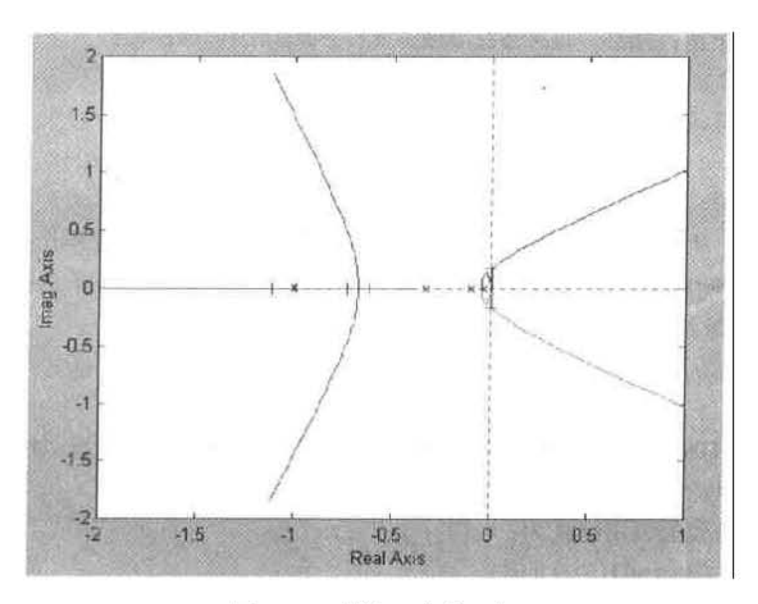


图 6-39　系统闭环根轨迹图

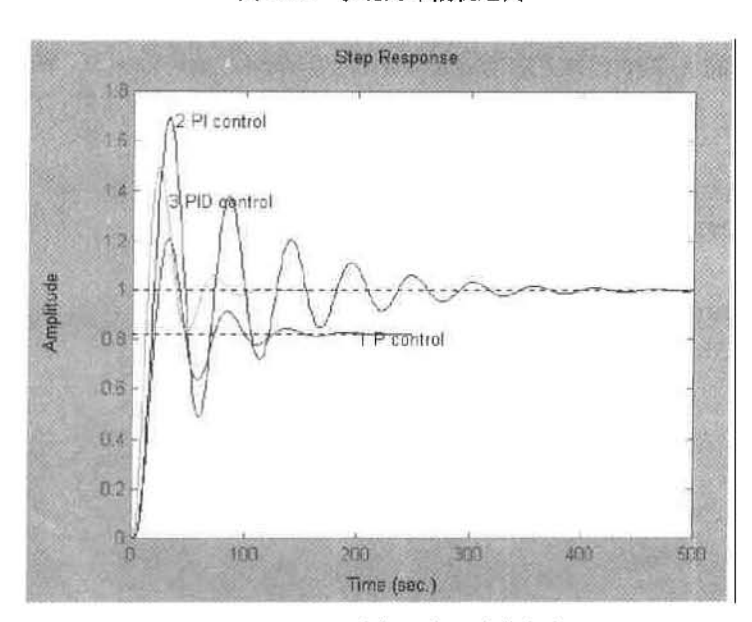


图 6-40　PID 三种校正阶跃给定响应

计算出三种 P、PI、PID 校正时校正器的参数分别为：

①P 校正器：$Gc1 = Kp1 = 0.4682$

②PI 校正器：$Kp2 = 0.5203$；$Ti2 = 32.4395$

```
Transfer function:
16.88 s + 0.5203
----------------
   32.44 s
```

③PID 校正器：*Kp*3 = 0.6861；*Ti*3 = 19.0821；*Td*3 = 4.7705

```
Transfer function:
62.46 s^2 + 13.09 s + 0.6861
----------------------------
   19.08 s
```

从图 6-40 所示的稳定边界法计算的 P、PI、PID 校正阶跃给定响应曲线看到，P 与 PI 校正的阶跃响应曲线上升的速度差不多快，PID 校正的最快；三条曲线有两个不同的终了值。超调量都较大，以 PI 校正的为最大。

附录：

【MATLAB FUNCTION PROGRAM 8】用 Ziegler –Nichols 整定公式计算系统 P、PI、PID 校正器参数的函数 zn01( )

zn01( ) 函数的调用格式为：

[*Gc,Kp,Ti,Td*]=zn01(*PID,vars*)

其中 *PID* 是校正器的类型，当 *PID*=1 时，为计算 P 调节器的参数；当 *PID*=2 时，为计算 PI 调节器的参数；当 *PID*=3 时，为计算 PID 调节器的参数。输入参量 *vars* 为带延迟-惯性环节模型的 K T τ 已知三参数：*K*=vars(1);*T*=vars(2);*tau*=vars(3)。输出参量 *Gc* 为校正器传递函数，K_p 为校正器的比例系数；*Ti* 为校正器的积分时间常数；*Td* 为校正器的微分时间常数。

```
function [Gc,Kp,Ti,Td]=zn01(PID,vars)
% MATLAB FUNCTION PROGRAM zn01.m
%
K=vars(1);T=vars(2);tau=vars(3);
Kp=[];Ti=[];Td=[];
if PID==1,
        Kp=T/(K*tau);
elseif PID==2,
        Kp=0.9*T/(K*tau);
        Ti=3.33*tau;
elseif PID==3,
        Kp=1.2*T/(K*tau);
        Ti=2*tau;Td=tau/2;
end
switch key
 case 1, Gc=Kp;
 case 2, Gc=tf([Kp*Ti Kp],[Ti 0]);
 case 3, nn=[Kp*Ti*Td Kp*Ti Kp];
         dd=[Ti 0];
         Gc=tf(nn,dd)
```

```
end
```

【MATLAB FUNCTION PROGRAM 9】非 $K\ T\ \tau$ 模型拟合成对应参数 K、T、τ 的函数 kttau( )

kttau( )函数的调用格式为：

[K,T,tau]=**kttau**(*G*)

其中输入参量 G 为模型的传递函数，输出参数 K、T、τ 为拟合成 $K\ T\ \tau$ 模型的参数。

```
function [K,T,tau]=kttau(G)
% MATLAB FUNCTION PROGRAM kttau.m
%
K=dcgain(G);
[Kc,Pm,Wcg,Wcp]=margin(G);
tau=1.6*pi/(3*Wcg);T=0.5*Kc*K*tau;
ktt=0;
if finite(Kc),
  x0=[tau;T];
  while ktt==0
        ww1=Wcg*x0(1);ww2=Wcg*x0(2);
        FF=[K*Kc*(cos(ww1)-ww2*sin(ww1))+1+ww2^2;sin(ww1)+ww2*cos(ww1)];
        J=[-K*Kc*Wcg*sin(ww1)-K*Kc*Wcg*ww2*cos(ww1),...
            -K*Kc*Wcg*sin(ww1)+2*Wcg*ww2;
            Wcg*cos(ww1)-Wcg*ww2*sin(ww1),Wcg*cos(ww1)];
        x1=x0-inv(J)*FF;
        if norm(x1-x0)<1e-8,
            ktt=1;
        else,x0=x1;
        end
        tau=x0(1);T=x0(2);
  end
end
```

【MATLAB FUNCTION PROGRAM 10】用 Cohen-Coon 整定公式计算系统 P、PI、PID 校正器的参数的函数 cc01( )

cc01( )函数的调用格式为：

[Gc,Kp,Ti,Td]=**cc01**(*PID,vars*)

其中 *PID* 是校正器的类型，当 *PID*=1 时，为计算 P 调节器的参数；当 *PID*=2 时，为计算 PI 调节器的参数；当 *PID*=3 时，为计算 PD 调节器的参数；当 *PID*=4 时，为计算 PID 调节器的参数。输入参量 *vars* 为带延迟-惯性环节模型的 $K\ T\ \tau$ 已知三参数：*K*=vars(1)；*T*=vars(2)；*tau*=vars(3)。输出参量 *Gc* 为校正器传递函数，*Kp* 为校正器的比例系数；*Ti* 为校正器的积分时间常数；*Td* 为校正器的微分时间常数。

```
function [Gc,Kp,Ti,Td]=cc01(PID,vars)
```

```
% MATLAB FUNCTION PROGRAM cc01.m
%
K=vars(1);T=vars(2);tau=vars(3);
Kp=[];Ti=[];Td=[];
if PID==1,
        Kp=[(T/tau)+0.333]/K;
elseif PID==2,
        Kp=[0.9*(T/tau)+0.082]/K;
        Ti=T*[3.33*(tau/T)+0.3*(tau/T)^2]/[1+2.2*(tau/T)];
elseif PID==3,
        Kp=[1.24*(T/tau)+0.1612]/K;
        Td=T*[0.27(tau/T)]/[1+0.13(tau/T)];
elseif PID==4,
        Kp=[1.35*(T/tau)+0.27]/K;
        Ti=T*[2.5*(tau/T)+0.5*(tau/T)^2]/[1+0.6*(tau/T)];
        Td=T*[0.37(tau/T)]/[1+0.2(tau/T)];
end
switch PID
  case 1, Gc=Kp;
  case 2, Gc=tf([Kp*Ti Kp],[Ti 0]);
  case 3, Gc=tf([Kp*Td Kp],1);
  case 4, nn=[Kp*Ti*Td Kp*Ti Kp];
         dd=[Ti 0];
         Gc=tf(nn,dd)
end
```

【MATLAB FUNCTION PROGRAM 11】用积分值最小准则来计算系统 P、PI、PID 校正器的参数的函数 inmin01( )

inmin01( )函数的调用格式为：

[*Gc*,*Kp*,*Ti*,*Td*]=inmin01(*PID*,*vars*)

其中 *PID* 是校正器的类型，当 *PID*=1 时，为计算 P 调节器的参数；当 *PID*=2 时，为计算 PI 调节器的参数；当 *PID*=3 时，为计算 PID 调节器的参数。输入参量 *vars* 为带延迟-惯性环节模型的 K T τ 已知三参数：*K*=vars(1)；*T*=vars(2)；*tau*=vars(3)。输出参量 *Gc* 为校正器传递函数，*Kp* 为校正器的比例系数；*Ti* 为校正器的积分时间常数；*Td* 为校正器的微分时间常数。

```
function [Gc,Kp,Ti,Td]=inmin01(PID,vars)
% MATLAB FUNCTION PROGRAM inmin01.m
%
K=vars(1);T=vars(2);tau=vars(3);iC=vars(4);
Kp=[];Ti=[];Td=[];
```

```
    if PID==1,
          ABtab=[0.902, 1.414, 0.904;
               -0.985,-0.917,-1.084];
    elseif PID==2,
          ABtab=[0.984, 1.305, 0.859;
               -0.986,-0.959,-0.977;
               0.608, 0.492, 0.674;
               -0.707,-0.739,-0.680];
    elseif PID==3,
          ABtab=[1.435, 1.495, 1.357;
               -0.921,-0.945,-0.947;
                0.878, 1.101, 0.842;
               -0.749,-0.771,-0.738;
                0.482, 0.560, 0.381;
                1.137, 1.006, 0.995];
    end
    ii=0;
    tt=tau/T;
    if PID==1,
       a1=ABtab(1,ii+iC);
       b1=ABtab(2,ii-iC);
       Kp=(a1*(tt^b1))/K;
    elseif PID==2,
       a1=ABtab(1,ii+iC);b1=ABtab(2,ii+iC);
       a2=ABtab(3,ii+iC);b2=ABtab(4,ii+iC);
       Kp=(a1*(tt^b1))/K;Ti=T/(a2*(tt^b2));
    elseif PID==3,
       a1=ABtab(1,ii+iC);b1=ABtab(2,ii+iC);
       a2=ABtab(3,ii+iC);b2=ABtab(4,ii+iC);
       a3=ABtab(5,ii+iC);b3=ABtab(6,ii+iC);
       Kp=(a1*(tt^b1))/K;Ti=T/(a2*(tt^b2));
       Td=T*a3*(tt^b3);
    end
if PID==1
     Gc=Kp;
elseif PID==2
      nn=[Kp*Ti Kp];
      dd=[Ti 0];
      Gc=tf(nn,dd);
```

```
elseif PID==3
      nn=[Kp*Ti*Td Kp*Ti Kp];
      dd=[Ti 0];
      Gc=tf(nn,dd);
end
```

【MATLAB FUNCTION PROGRAM 12】用稳定边界法整定公式计算系统 P、PI、PID 校正器的参数的函数 zn02( )

zn02( )函数的调用格式为：

[*Gc*,*Kp*,*Ti*,*Td*]=zn02(*PID*,*vars*)

其中 *PID* 是校正器的类型，当 *PID*=1 时，为计算 P 调节器的参数；当 *PID*=2 时，为计算 PI 调节器的参数；当 *PID*=3 时，为计算 PID 调节器的参数。输入参量 *vars* 为带延迟-惯性环节模型的 *K* *T* τ 已知三参数：*K*=vars(1)；*T*=vars(2)；*tau*=vars(3)。输出参量 *Gc* 为校正器传递函数，*Kp* 为校正器的比例系数；*Ti* 为校正器的积分时间常数；*Td* 为校正器的微分时间常数。

```
function [Gc,Kp,Ti,Td]=zn02(PID,G)
% MATLAB FUNCTION PROGRAM zn02.m
%
Kp=[];Ti=[];Td=[];
rlocus(G);
[km,pole]=rlocfind(G)
keyboard
wm=imag(pole(4))
if PID==1,
        Kp=0.5*km;;
elseif PID==2,
        Kp=0.455*km;
        Ti=0.85*2*pi/wm;
elseif PID==3,
        Kp=0.6*km;
        Ti=0.5*2*pi/wm;
        Td=0.125*2*pi/wm;
end
switch PID
  case 1, Gc=Kp
  case 2, Gc=tf([Kp*Ti Kp],[Ti 0])
  case 3, nn=[Kp*Ti*Td Kp*Ti Kp];
        dd=[Ti 0];
        Gc=tf(nn,dd)
end
```

第 7 章　状态空间分析的 MATLAB 实现

状态空间模型是控制系统最新与最科学的描述方法，它能够完全表达系统的全部状态与性能。它不但能描述线性系统，而且也能描述非线性系统与时变系统。它既能描述单输入单输出系统，也能描述多输入多输出系统。状态空间模型是现代控制理论的基础。本章共 6 节：7.1 节首先简要复习控制系统状态空间分析的几个基本概念；7.2 节介绍系统状态空间表达式及其状态方程的解；7.3 节介绍系统状态方程的线性变换；7.4 节介绍线性系统的可控性与可观性；7.5 节介绍系统状态反馈与状态观测器；7.6 节介绍系统稳定性分析的 MATLAB 计算及仿真。状态空间模型都是用矩阵表示的，而 MATLAB 是矩阵实验室，是以矩阵作为基本运算单元的，所以使用 MATLAB 工具进行状态空间分析的计算与仿真是最佳选择。

7.1　控制系统状态空间分析的几个基本概念

在经典控制理论中，常用传递函数这个数学模型对于单输入-单输出系统进行计算、分析及设计，这是一种行之有效的方法。但是传递函数只能描述系统输出量与输入量之间的外部关系，而对于系统内部各变量的状况则无从知晓。其次，传递函数数学模型是基于零初始条件下的拉普拉斯变换，因此，它只是系统的零状态模型，不能完全反映系统零输入响应的动态特征。这种模型没有包含系统的全部信息，是不完整与不全面的。

在现代控制理论中，用状态变量法来描述系统，这时控制系统是用一阶矩阵向量微分方程来描述的。之所以采用矩阵表示法是为了使描述系统的数学表达式简捷而明了，方便而高效，并且容易用计算机求解，同时也为多变量系统与时变系统的分析与研究提供有力的工具。

为介绍状态空间分析的 MATLAB 计算与仿真，首先复习自动控制基础课程中学过的几个重要基本概念。

1. 输入

即由外部施加到系统上的全部激励作用。

2. 输出

即能够从外部量测到的来自系统的信息。

3. 状态

动力学系统的状态可以定义为信息的集合。在未来已知系统外部输入的条件下，这些信息对于确定系统未来的行为是必要且充分的。因此，动力学系统在时间 t 的状态是

由 t_0 时的状态和 $t \geq t_0$ 时的输入唯一确定，而它是与 t_0 前的状态与输入无关的。

4. 状态变量

动力学系统的状态变量是确定动力学系统状态的最小一组变量。如果以最少的 n 个变量 $x_1(t)$、$x_2(t)$……、$x_n(t)$ 就能够完全描述动力学系统的行为（即当 $t \geq t_0$ 时输入和在 $t=t_0$ 时的初始状态给定后，系统的状态将完全可以被确定），那么这样的 n 个变量 $x_1(t)$、$x_2(t)$……，$x_n(t)$ 就是系统的一组状态变量。但对于该系统而言，这一组状态变量不是唯一的。系统的状态可由某一组状态变量来描述，也可由其他组的状态变量来描述，它们都能表达系统的状态与性能。

5. 状态向量

如果完全描述一个给定系统的动态行为需要 n 个状态变量，那么可将这些状态变量看作是向量 $\boldsymbol{x}(t)$ 的各个分量，即

$$\boldsymbol{x}(t)=\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{7-1}$$

$\boldsymbol{x}(t)$ 就叫做状态向量。

6. 状态空间

以各状态变量作为坐标轴所组成的 n 维空间叫做状态空间。在特定的时刻 t，状态向量则可用状态空间中的一个点来表示。

7. 状态轨线

状态向量 $\boldsymbol{x}(t)$ 在状态空间中随时间 t 变化的轨迹。

8. 连续时间系统

状态向量 $\boldsymbol{x}(t)$ 的定义域为某时间域$[t_0, t_f]$内的一切实数。

9. 离散时间系统

状态向量 $\boldsymbol{x}(t)$ 的自变量时间 t 只能取到某实数域内的离散值。

10. 状态方程

描述系统状态变量与输入变量之间关系的一阶微分方程组（连续时间系统）或一阶差分方程组（离散时间系统）。其一般形式：

$$\dot{\boldsymbol{x}}=f(\boldsymbol{x},\boldsymbol{u},t) \tag{7-2}$$

或

$$\boldsymbol{x}(t_{k+1})=f[\boldsymbol{x}(t_k),\boldsymbol{u}(t_k),t_k] \tag{7-3}$$

状态方程表征了系统由输入所引起的内部状态的变化。

11. 输出方程

描述系统输出变量与输入变量及状态变量之间函数关系的代数方程。其一般形式为：

$$\boldsymbol{y}(t)=g[\boldsymbol{x}(t),\boldsymbol{u}(t),t] \tag{7-4}$$

输出方程表征了系统内部状态的变化和输入所引起系统输出的变化，它是一个变换过程。

12. 状态空间表达式

状态方程与输出方程联立起来，构成一个对动态系统的完整描述，总称为系统的状态空间表达式，又叫做动态方程。其一般形式为：

$$\begin{cases} \dot{\boldsymbol{x}}(t) = f[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \\ \boldsymbol{y}(t) = g[\boldsymbol{x}(t), \boldsymbol{u}(t), t] \end{cases} \tag{7-5}$$

或

$$\begin{cases} \boldsymbol{x}(t_{k+1}) = f[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k] \\ \boldsymbol{y}(t_k) = g[\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k] \end{cases} \tag{7-6}$$

13. 线性系统

若在系统的状态空间表达式中，f 与 g 都是线性函数，则称系统为线性系统，否则为非线性系统。

14. 线性系统状态空间表达式

线性系统的状态方程是一阶向量线性微分方程或一阶向量线性差分方程；输出方程是向量代数方程。对于连续时间系统它的一般形式为：

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}(t)\boldsymbol{x}(t) + \boldsymbol{D}(t)\boldsymbol{u}(t) \end{cases} \tag{7-7}$$

而对于离散时间系统，由于在实践中常取 $t_k = kT$（T 叫做该离散时间系统的采样周期），故其状态空间表达式的一般形式可为：

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{F}(k)\boldsymbol{x}(k) + \boldsymbol{G}(k)\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \boldsymbol{C}(k)\boldsymbol{x}(k) + \boldsymbol{D}(k)\boldsymbol{u}(k) \end{cases} \tag{7-8}$$

一般情况下，若状态 $\boldsymbol{x}$、输入 $\boldsymbol{u}$、输出 $\boldsymbol{y}$ 的维数分别为 n、r、m，则称 $n \times n$ 矩阵 $\boldsymbol{A}(t)$ 与 $\boldsymbol{F}(k)$ 为系统矩阵或状态矩阵；称 $n \times r$ 矩阵 $\boldsymbol{B}(t)$ 与 $\boldsymbol{G}(k)$ 为控制矩阵或输入矩阵；称 $m \times n$ 矩阵 $\boldsymbol{C}(t)$ 与 $\boldsymbol{C}(k)$ 为观测矩阵或输出矩阵；称 $m \times r$ 矩阵 $\boldsymbol{D}(t)$ 与 $\boldsymbol{D}(k)$ 为输入输出矩阵。

15. 线性系统的动态结构图

线性连续时间系统的动态结构图如图 7-1 所示。类似地，线性离散时间系统的动态结构图如图 7-2 所示。

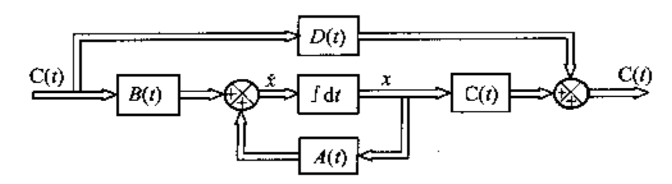


图 7-1　线性连续时间系统的动态结构图

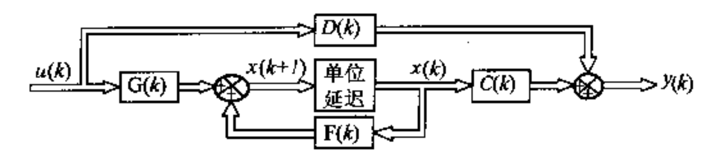


图 7-2　线性离散时间系统的动态结构图

16. 线性定常系统

在线性系统状态空间表达式中，若系数矩阵$A(t)$、$B(t)$、$C(t)$、$D(t)$或$F(k)$、$G(k)$、$C(k)$、$D(k)$的各元素都是常数，则称该系统为线性定常系统，否则叫线性时变系统。线性定常系统的一般形式为：

$$\begin{cases}\dot{x}(t)=Ax(t)+Bu(t)\\ y(t)=Cx(t)+Du(t)\end{cases}\tag{7-9}$$

或

$$\begin{cases}x(k+1)=Fx(k)+Gu(k)\\ y(k)=Cx(k)+Du(k)\end{cases}\tag{7-10}$$

7.2 系统状态空间表达式及其状态方程的解

7.2.1 状态空间表达式的建立

控制系统状态空间表达式的建立主要有三种方法：第一种是直接根据控制系统的工作原理建立相应的微分方程或差分方程，再将其整理、规范化而得；第二种方法是由控制系统的结构方框图建立系统状态空间表达式；第三种方法是由已知系统的某种数学模型转化而得。以下举例说明系统状态空间表达式的建立，【例 7-1】是第一种方法，【例 7-2】是第二种方法，【例 7-3】与【例 7-4】是第三种方法。

【例 7-1】对于电枢电压控制的它激直流电动机，试建立其状态空间表达式。

【解】

（1）它激直流电动机系统的微分方程

电枢电压控制的恒定激磁它激直流电动机电路如图 7-3 所示，电枢电压与负载转矩分别是系统的给定输入与扰动输入。

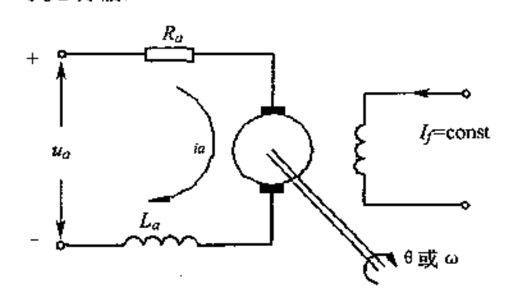


图 7-3　它激直流电动机电路

①根据基尔霍夫第二定理有电枢回路方程：

$$L_a\frac{\mathrm{d}i_a}{\mathrm{d}t}=u_a-R_ai_a-K_e\omega\tag{7-11}$$

②根据电机及拖动装置的动力学方程（考虑粘性摩擦转矩）有：

$$J\frac{\mathrm{d}\omega}{\mathrm{d}t}=M_D-M_L-f\omega\tag{7-12}$$

$$M_D = K_M i_a \tag{7-13}$$

（2）整理规范化为状态方程

对式（7-11）~式（7-13）做恒等变换，则有：

$$\begin{cases} \dot{i}_a = -\dfrac{R_a}{L_a} i_a - \dfrac{K_e}{L_a}\omega + \dfrac{1}{L_a} u_a \\ \dot{\omega} = \dfrac{K_M}{J} i_a - \dfrac{f}{J}\omega - \dfrac{1}{J} M_L \end{cases} \tag{7-14}$$

将式（7-14）写成向量与矩阵形式：

$$\boldsymbol{x} = \begin{bmatrix} i_a \\ \omega \end{bmatrix}$$

$$\boldsymbol{u} = \begin{bmatrix} u_a \\ M_L \end{bmatrix}$$

可得系统状态空间表达式为：

$$\begin{cases} \dot{\boldsymbol{x}} = \begin{bmatrix} -\dfrac{R_a}{L_a} & -\dfrac{K_e}{L_a} \\ \dfrac{K_M}{J} & -\dfrac{f}{J} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \dfrac{1}{L_a} & 0 \\ 0 & -\dfrac{1}{J} \end{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} = \begin{bmatrix} 0 & 1 \end{bmatrix} x \end{cases}$$

其系统矩阵 $\boldsymbol{A}$ 与控制矩阵 $\boldsymbol{B}$ 分别为：

$$\begin{cases} \boldsymbol{A} = \begin{bmatrix} -\dfrac{R_a}{L_a} & -\dfrac{K_e}{L_a} \\ \dfrac{K_M}{J} & -\dfrac{f}{J} \end{bmatrix} \\ \boldsymbol{B} = \begin{bmatrix} \dfrac{1}{L_a} & 0 \\ 0 & -\dfrac{1}{J} \end{bmatrix} \end{cases}$$

【例 7-2】已知控制系统方框图如 图 7-4 所示，试求其状态空间表达式。

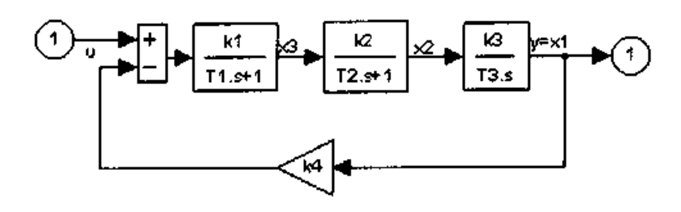


图 7-4　控制系统方框图

【解】

由系统方框图 7-4 可得模拟结构图 7-5。由模拟结构图 7-5，逐级写出其对应的关系式如下：

$$\dot{x}_1 = \frac{k_3}{T_3} x_2 \tag{7-15}$$

$$\dot{x}_2 = -\frac{1}{T_2} x_2 + \frac{k_2}{T_2} x_3 \tag{7-16}$$

$$\dot{x}_3 = -\frac{k_1 k_4}{T_1} x_1 - \frac{1}{T_1} x_3 + \frac{k_1}{T_1} \boldsymbol{u} \tag{7-17}$$

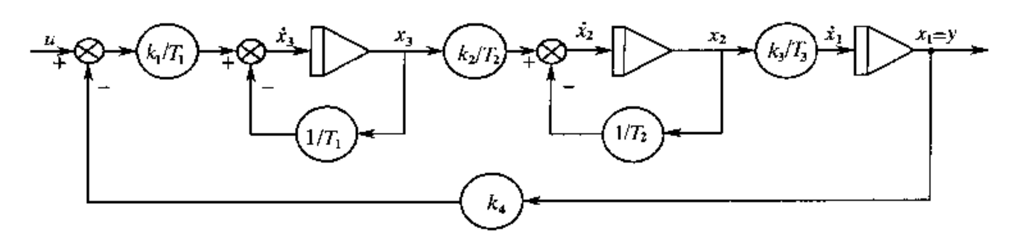


图 7-5　系统模拟结构图

式（7-15）~式（7-17）为系统的状态方程。由图 7-4，有系统输出方程

$$y = x_1$$

写出向量矩阵形式，则系统的状态空间表达方式为

$$\begin{cases} \dot{\boldsymbol{x}} = \begin{bmatrix} 0 & \dfrac{k_3}{T_3} & 0 \\ 0 & -\dfrac{1}{T_2} & \dfrac{k_2}{T_2} \\ -\dfrac{k_1 k_4}{T_1} & 0 & -\dfrac{1}{T_1} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{\boldsymbol{k}_1}{\boldsymbol{T}_1} \end{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} = [1 \quad 0 \quad 0] \boldsymbol{x} \end{cases}$$

式中　$\boldsymbol{x}^T = [x_1 \quad x_2 \quad x_3]$

【例 7-3】已知连续系统的零极点增益模型为：

$$\boldsymbol{H}(s) = \frac{10(s+5)}{(s+1)(s+3)(s+8)}$$

求系统的状态空间空间表达式。

【解】

为求系统的状态空间模型，特给出以下程序 L7203.m 来实现，程序中调用了建立与转换系统模型的函数 ss（）。

```
% MATLAB PRAGRAM L7203.m
%
k=10;z=-5;p=[-1,-3,-8];
sys=zpk(z,p,k);ss(sys)
```

程序运行后可得：

```
a =
```

```
              x1          x2          x3
      x1      -8          0.89443     2.2361
      x2      0           -4          -1.5
      x3      0           2           0
b =
              u1
      x1      0
      x2      3.5355
      x3      0
c =
              x1          x2          x3
      y1      3.1623      0           0
d =
              u1
      y1      0
Continuous-time model.
```

根据计算的系数矩阵可得系统的状态空间表达式为：

$$\begin{cases}\dot{x}=\begin{bmatrix}-8 & 0.89443 & 2.2361\\ 0 & -4 & -1.5\\ 0 & 2 & 0\end{bmatrix}x+\begin{bmatrix}0\\ 3.5355\\ 0\end{bmatrix}u\\ y=\begin{bmatrix}3.1623 & 0 & 0\end{bmatrix}x\end{cases}$$

【例 7-4】已知连续系统的传递函数模型为：

$$G(s)=\frac{1}{s^2+3s+2}$$

求系统的状态空间表达式。

【解】

为求系统的状态空间表达式，给出以下程序 L7204.m。

```
% MATLAB PROGRAM L7204.m
%
num=[0,0,1];den=[1,3,2];s1=tf(num,den);
ss(s1)
```

程序运行结果得到：

```
a =
              x1          x2
      x1      -3          -1
      x2       2           0
b =
              u1
```

```
          x1          0.5
          x2            0
c =
                      x1          x2
          y1            0           1
d =
                      u1
          y1            0
```

Continuous-time model.

由以上计算的数据可得系统的状态空间表达式为:

$$\begin{cases}\dot{x}=\begin{bmatrix}-3 & -1\\ 2 & 0\end{bmatrix}x+\begin{bmatrix}0.5\\ 0\end{bmatrix}u\\ y=\begin{bmatrix}0 & 1\end{bmatrix}x\end{cases}$$

7.2.2 系统状态方程的解

控制系统的数学模型建立以后，只有求得了系统状态方程的解，才能确定系统的行为，完成系统分析与设计的任务。

考察非齐次状态方程

$$\dot{\boldsymbol{x}}=\boldsymbol{Ax}+\boldsymbol{Bu} \tag{7-18}$$

式中 $\boldsymbol{x}$ 为 n 维向量，$\boldsymbol{u}$ 为 r 维向量，$\boldsymbol{A}$ 为 $n\times n$ 常数矩阵，$\boldsymbol{B}$ 为 $n\times r$ 常数矩阵。把式（7-18）改写成

$$\dot{\boldsymbol{x}}(t)-\boldsymbol{Ax}(t)=\boldsymbol{Bu}(t)$$

以 e^{-At} 左乘等式两边有:

$$\mathrm{e}^{-At}\left[\dot{\boldsymbol{x}}(t)-\boldsymbol{Ax}(t)\right]=\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathrm{e}^{-At}\boldsymbol{x}(t)\right]=\mathrm{e}^{-At}\boldsymbol{Bu}(t)$$

在 0 到 t 之间对上式进行积分，可得:

$$\mathrm{e}^{-At}\boldsymbol{x}(t)=\boldsymbol{x}(0)+\int_0^t\mathrm{e}^{-A\tau}\boldsymbol{Bu}(\tau)\mathrm{d}\tau$$

再以 e^{At} 左乘上式两边，即得到状态方程的解:

$$\boldsymbol{x}(t)=\mathrm{e}^{At}\boldsymbol{x}(0)+\int_0^t\mathrm{e}^{A(t-\tau)}\boldsymbol{Bu}(\tau)\mathrm{d}\tau \tag{7-19}$$

或

$$\boldsymbol{x}(t)=\boldsymbol{\Phi}(t)\boldsymbol{x}(0)+\int_0^t\boldsymbol{\Phi}(t-\tau)\boldsymbol{Bu}(\tau)\mathrm{d}\tau \tag{7-20}$$

若状态方程是齐次的，即有:

$$\dot{\boldsymbol{x}}=\boldsymbol{Ax} \tag{7-21}$$

其解为

$$\boldsymbol{x}(t)=\mathrm{e}^{At}\boldsymbol{x}(0) \tag{7-22}$$

或
$$x(t)=\Phi(t)x(0) \tag{7-23}$$

在以上的式（7-19）、（7-20）、（7-22）与（7-23）中，均假定初始时刻为零。如果初始时刻为 t_0，则 $x(0)$ 应改为 $x(t_0)$，$\Phi(t)=e^{At}$ 应改为 $\Phi(t-t_0)=e^{A(t-t_0)}$，积分式的下限 0 应改为 t_0。

根据幂级数展开公式有：

$$\Phi(t)=e^{At}=I+At+\frac{1}{2!}A^2t^2+\cdots+\frac{1}{k!}A^kt^k+\cdots=L^{-1}\left[(sI-A)^{-1}\right] \tag{7-24}$$

$\Phi(t)$叫做控制系统的状态转移矩阵（又称为矩阵指数函数）。

【例 7-5】已知控制系统状态方程为：

$$\dot{x}=\begin{bmatrix}-2 & \sqrt{2}\\ \sqrt{2} & -1\end{bmatrix}x$$

用拉氏变换法求控制系统的状态转移矩阵 $\Phi(t)$。

【解】

（1）求特征矩阵 $[sI-A]$

$$A=\begin{bmatrix}-2 & \sqrt{2}\\ \sqrt{2} & -1\end{bmatrix}$$

$$[sI-A]=\begin{bmatrix}s+2 & -\sqrt{2}\\ -\sqrt{2} & s+1\end{bmatrix}$$

（2）求矩阵 $[sI-A]$ 的行列式

$$|sI-A|=(s+2)(s+1)-2=s(s+3)$$

（3）求矩阵 $[sI-A]$ 的伴随矩阵

$$\mathrm{adj}[sI-A]=\begin{bmatrix}s+1 & \sqrt{2}\\ \sqrt{2} & s+2\end{bmatrix}$$

（4）求矩阵 $[sI-A]$ 的逆

$$[sI-A]^{-1}=\frac{1}{s(s+3)}\begin{bmatrix}s+1 & \sqrt{2}\\ \sqrt{2} & s+2\end{bmatrix}$$

（5）根据拉氏反变换的公式求状态转移矩阵

$$\Phi(t)=e^{At}=L^{-1}\left\{[sI-A]^{-1}\right\}=\begin{bmatrix}\frac{1}{3}\left(1+2e^{-3t}\right) & \frac{\sqrt{2}}{3}\left(1-e^{-3t}\right)\\ \frac{\sqrt{2}}{3}\left(1-e^{-3t}\right) & \frac{1}{3}\left(2+e^{-3t}\right)\end{bmatrix}$$

【例 7-6】求控制系统 $\frac{dx}{dt}=Ax$ 的 $x(0)=[1;1;1;]$ 的解。

$$A=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -6 & -11 & -6\end{bmatrix}$$

求控制系统的状态转移矩阵$\boldsymbol{\Phi}(t)$。

【解】

由式（7-21）、式（7-22）与式（7-23）得：$\boldsymbol{x}(t)=e^{At}\boldsymbol{x}(0)=\boldsymbol{\Phi}(t)\boldsymbol{x}(0)$ 而系统的状态转移矩阵式（7-24）：

$$\boldsymbol{\Phi}(t)=\mathrm{e}^{At}=\boldsymbol{I}+\boldsymbol{A}t+\frac{1}{2!}\boldsymbol{A}^2t^2+\cdots+\frac{1}{k!}\boldsymbol{A}^kt^k+\cdots$$

即为求矩阵的指数函数。根据计算矩阵指数的函数 expm （ ）,特给出以下程序 L7206.m 实现求解。

```
% MATLAB PROGRAM L7206.m
%
a =[0 1 0;0 0 1;-6 -11 -6];
x0=[1;1;1];
t=0:0.1:10;
for i=1:length (t)
   x (:,i) =expm (a*t (i)) *x0;
end
plot3 (x (1,:) ,x (2,:) ,x (3,:)) ;
grid on
```

程序执行结果如图 7-6 所示。

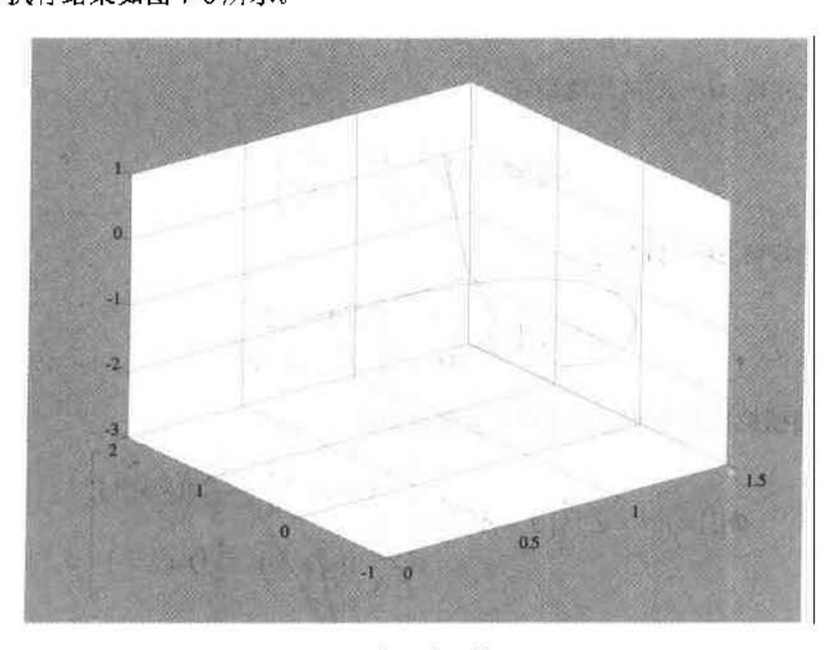


图 7-6　矩阵指数函数三维图

【例 7-7】已知控制系统为：

$$\dot{x}=\begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\-6 & -11 & -6\end{bmatrix}x+\begin{bmatrix}0\\0\\1\end{bmatrix}u$$

求控制系统的特征方程、特征值及特征向量。

【解】

（1）求特征方程

因为

$$A=\begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\-6 & -11 & -6\end{bmatrix}$$

则有：

$$|sI-A|=\begin{bmatrix}s & -1 & 0\\0 & s & -1\\6 & 11 & s+6\end{bmatrix}=0$$

即

$$|sI-A|=s^3+6s^2+11s+6=0$$

用以下 MATLAB 语句分解因式：

```
p=[1 6 11 6];roots (p)
ans =
   -3.0000
   -2.0000
   -1.0000
```

所以特征方程为：

$$|sI-A|=s^3+6s^2+11s+6=(s+1)(s+2)(s+3)=0$$

（2）求特征值及特征向量

以下 MATLAB 语句为用函数 eig（ ）求矩阵 A 的特征向量与特征值。

```
A=[0 1 0;0 0 1;-6 -11 -6];
[V,D]=eig (A)
```

得矩阵 A 的特征向量矩阵 V：

```
V =
   -0.5774    0.2182   -0.1048
    0.5774   -0.4364    0.3145
   -0.5774    0.8729   -0.9435
D =
   -1.0000         0         0
         0   -2.0000         0
         0         0   -3.0000
```

（3）验算矩阵 **A** 特征向量与特征值的求解是否正确

A*V=V*D，即两者结果相同，表明特征向量矩阵与特征值矩阵计算正确。

【例 7-8】已知控制系统

$$A=\begin{bmatrix}2 & 1 & -1\\ 1 & 2 & -1\\ -1 & -1 & 2\end{bmatrix}$$

求控制系统的特征方程、特征值及特征向量。

【解】

（1）求特征方程

$$|s\boldsymbol{I}-\boldsymbol{A}|=\begin{vmatrix}s-2 & -1 & 1\\ -1 & s-2 & 1\\ 1 & 1 & s-2\end{vmatrix}=s^3-6s^2+9s-4=0$$

用以下 MATLAB 语句分解因式：

```
p=[1 -6 9 -4];r=roots(p)
r =
   4.0000
   1.0000 + 0.0000i
   1.0000 - 0.0000i
```

即有：

$$|s\boldsymbol{I}-\boldsymbol{A}|=s^3-6s^2+9s-4=(s-4)(s-1)(s-1)=0$$

（2）求特征值及特征向量

```
A=[2 1 -1;1 2 -1;-1 -1 2];
[V,D]=eig(A)
V =
   -0.8054    0.1340   -0.5774
    0.5187    0.6305   -0.5774
   -0.2867    0.7645    0.5774
D =
    1.0000         0         0
         0    1.0000         0
         0         0    4.0000
```

（3）验算矩阵 **A** 特征向量与特征值的求解是否正确

```
A*V
ans =
   -0.8054    0.1340   -2.3094
    0.5187    0.6305   -2.3094
   -0.2867    0.7645    2.3094
```

```
V*D
ans =
   -0.8054    0.1340   -2.3094
    0.5187    0.6305   -2.3094
   -0.2867    0.7645    2.3094
```

A*V=V*D，即两者结果相同，表明特征向量矩阵与特征值矩阵计算正确。

（4）特征向量矩阵 ***V*** 可将矩阵 ***A*** 转化为对角矩阵

```
A=[2 1 -1;1 2 -1;-1 -1 2];
[V,D]=eig (A) ;
V1=inv (V)
V1 =
   -0.8054    0.5187   -0.2867
    0.1340    0.6305    0.7645
   -0.5774   -0.5774    0.5774
V1*A*V
ans =
    1.0000   -0.0000   -0.0000
   -0.0000    1.0000   -0.0000
         0   -0.0000    4.0000
```

计算结果说明特征向量矩阵 ***V*** 能将矩阵 ***A*** 对角化。

【例 7-9】已知系统状态方程为：

$$\dot{\boldsymbol{x}}=\begin{bmatrix}0 & 1\\ -2 & -3\end{bmatrix}\boldsymbol{x}$$

当 $\boldsymbol{x}_1(0)=1,\boldsymbol{x}_2(0)=0$ 时，试求状态方程之解。

【解】

（1）求特征矩阵及逆

$$[s\boldsymbol{I}-\boldsymbol{A}]=\begin{bmatrix}s & 0\\ 0 & s\end{bmatrix}-\begin{bmatrix}0 & 1\\ -2 & -3\end{bmatrix}=\begin{bmatrix}s & -1\\ 2 & s+3\end{bmatrix}=s^2+3s+2$$

$$[s\boldsymbol{I}-\boldsymbol{A}]^{-1}=\frac{\operatorname{adj}[s\boldsymbol{I}-\boldsymbol{A}]}{|s\boldsymbol{I}-\boldsymbol{A}|}=\frac{1}{(s+1)(s+2s)}\begin{bmatrix}s+3 & 1\\ -2 & s\end{bmatrix}=$$

$$\begin{bmatrix}\dfrac{s+3}{(s+1)(s+2)} & \dfrac{1}{(s+1)(s+2)}\\ \dfrac{-2}{(s+1)(s+2)} & \dfrac{s}{(s+1)(s+2)}\end{bmatrix}=\begin{bmatrix}\dfrac{2}{s+1}-\dfrac{1}{s+2} & \dfrac{1}{s+1}-\dfrac{1}{s+2}\\ -\dfrac{2}{s+1}+\dfrac{2}{s+2} & -\dfrac{1}{s+1}+\dfrac{2}{s+2}\end{bmatrix}$$

（2）用拉氏反变换法求状态转移矩阵

$$\boldsymbol{\Phi}(t)=L^{-1}\{[sI-A]^{-1}\}=\begin{bmatrix}(2e^{-t}-e^{-2t}) & (e^{-t}-e^{-2t})\\(-2e^{-t}+2e^{-2t}) & (-e^{-t}+2e^{-2t})\end{bmatrix}$$

（3）根据矩阵乘法运算规则可得状态方程的解为：

$$\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix}=\boldsymbol{\Phi}(t)\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}=\begin{bmatrix}(2e^{-t}-e^{-2t}) & (e^{-t}-e^{-2t})\\(-2e^{-t}+2e^{-2t}) & (-e^{-t}+2e^{-2t})\end{bmatrix}\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}$$

已知 $x_1(0)=1, x_2(0)=0$, 则有解：

$$x_1(t)=2e^{-t}-e^{-2t}$$

$$x_2(t)=2e^{-t}+2e^{-2t}$$

【例 7-10】已知系统状态方程为：

$$\dot{\boldsymbol{x}}=\begin{bmatrix}0 & 1\\-2 & -3\end{bmatrix}\boldsymbol{x}+\begin{bmatrix}0\\1\end{bmatrix}u$$

当 $\boldsymbol{x}(0)=0$ 时，试求控制量为单位阶跃函数（u=1）时的系统状态的时间响应。

【解】

（1）由上题，系统状态转移矩阵

$$\boldsymbol{\Phi}(t)=\begin{bmatrix}(2e^{-t}-e^{-2t}) & (e^{-t}-e^{-2t})\\(-2e^{-t}+2e^{-2t}) & (-e^{-t}+2e^{-2t})\end{bmatrix}$$

（2）根据式（7-20），有状态方程的解

$$\boldsymbol{x}(t)=\boldsymbol{\Phi}(t)\boldsymbol{x}(0)+\int_0^t\boldsymbol{\Phi}(t-\tau)\boldsymbol{B}u(\tau)\mathrm{d}\tau=$$

$$\boldsymbol{\Phi}(t)\boldsymbol{x}(0)+\int_0^t\begin{bmatrix}2e^{-(t-\tau)}-e^{-2(t-\tau)} & e^{-(t-\tau)}-e^{-2(t-\tau)}\\-2e^{-(t-\tau)}+2e^{-2(t-\tau)} & -e^{-(t-\tau)}+2^{-2(t-\tau)}\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}[1]\mathrm{d}\tau$$

经计算可得：

$$\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix}=\begin{bmatrix}(2e^{-t}-e^{-2t}) & (e^{-t}-e^{-2t})\\(-2e^{-t}+2e^{-2t}) & (-e^{-t}+2e^{-2t})\end{bmatrix}\begin{bmatrix}x_1(0)\\x_2(0)\end{bmatrix}+\begin{bmatrix}\frac{1}{2}-e^{-t}+\frac{1}{2}e^{-2t}\\e^{-t}-e^{-2t}\end{bmatrix}$$

如果系统的初始状态为零状态，即 $\boldsymbol{x}(0)=0$，那么系统在单位阶跃信号输入下的时间响应为：

$$\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix}=\begin{bmatrix}\frac{1}{2}-e^{-t}+\frac{1}{2}e^{-2t}\\e^{-t}-e^{-2t}\end{bmatrix}$$

7.2.3 连续系统状态方程的离散化

在对连续系统进行实时计算机控制时，往往需要把连续系统转换成离散系统。此时数字机将参与自动控制，处理离散的数据。一方面数值是量化的，同时时间也是量化的。除此之外，数字机在对采样点上的数据进行处理之后，给出的控制量 $\boldsymbol{u}(t)$ 在采样间隔期间内将保持为常值，故 $\boldsymbol{u}(t)$ 为一阶梯信号：

$$\boldsymbol{u}[kT\leqslant t\leqslant(k+1)T]=\boldsymbol{u}(kT)\quad(k=0,1,2,\cdots)\tag{7-25}$$

式中 $\boldsymbol{T}$ 为采样周期。

自动控制系统通常是一个连续系统，它在阶梯信号 $\boldsymbol{u}(kT)$ 的作用下将产生连续响应。系统的连续状态变量只有在采样点上的数值 $\boldsymbol{x}(T)$ 、 $\boldsymbol{x}(2T)$ 、 $\boldsymbol{x}(3T)$ 、……才会被数字机当作有用信息来处理。

设线性定常系统的动态方程为：

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)$$

式中，状态 $\boldsymbol{x}$ 、输入 $\boldsymbol{u}$ 、输出 $\boldsymbol{y}$ ，矩阵 $\boldsymbol{A}$ 为系统矩阵或状态矩阵，矩阵 $\boldsymbol{B}$ 为控制矩阵或输入矩阵，矩阵 $\boldsymbol{C}$ 为观测矩阵或输出矩阵，矩阵 $\boldsymbol{D}$ 为输入输出矩阵。线性定常系统中 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}$ 均为常数阵。

如果知道在 kT 时刻连续系统的输入量为 $\boldsymbol{u}(kT)$，而且在 $kT\leqslant t\leqslant(k+1)T$ 的时间间隔内 $\boldsymbol{u}(kT)$ 为一常量。此外，在 kT 时刻的状态量 $\boldsymbol{x}(kT)$ 亦为已知的话，则可通过式（7-21）计算系统在 $\boldsymbol{u}(kT)$ 作用下的响应为：

$$\boldsymbol{x}(t)=\boldsymbol{\Phi}(t-kT)\boldsymbol{x}(kT)+\int_{kT}^{t}\boldsymbol{\Phi}(t-\tau)\boldsymbol{B}\boldsymbol{u}(\tau)\mathrm{d}\tau \tag{7-26}$$

欲求 $t=(k+1)T$ 时刻之状态量 $\boldsymbol{x}[(k+1)T]$，只要令上式中 $t=(k+1)T$ 即可，于是有：

$$\boldsymbol{x}[(k+1)T]=\boldsymbol{\Phi}(T)\boldsymbol{x}(kT)+\int_{kT}^{(k+1)T}\boldsymbol{\Phi}[(k+1)T-\tau]\boldsymbol{B}\mathrm{d}\tau\cdot\boldsymbol{u}(kT) \tag{7-27}$$

若令

$$G(T)=\int_{kT}^{(k+1)T}\boldsymbol{\Phi}[(k+1)T-\tau]\boldsymbol{B}\mathrm{d}\tau \tag{7-28}$$

则式（7-27）可写成：

$$\boldsymbol{x}[(k+1)T]=\boldsymbol{\Phi}(T)\boldsymbol{x}(kT)+\boldsymbol{G}(T)\boldsymbol{u}(kT) \tag{7-29}$$

此式即为连续系统的离散化状态方程，相应地有系统离散化的输出方程为：

$$\boldsymbol{y}(kT)=\boldsymbol{C}\boldsymbol{x}(kT)+\boldsymbol{D}\boldsymbol{u}(kT) \tag{7-30}$$

为了实现连续系统状态方程的离散化，在此介绍 MATLAB 的函数 c2d（ ），它的功能是将连续时间模型转换为离散时间模型。函数 c2d（ ）的调用格式为：

sysd=**c2d**（*sysc,Ts*）

sysd=**c2d**（*sysc,Ts,method*）

其中，输入参量 sysc 为连续时间模型对象；*Ts* 为采样周期，单位为秒；*sysd* 为带采样时间 *Ts* 的离散时间模型。method 用来指定离散化采用的方法：

'zoh' —— 采用零阶保持器；

'foh' —— 采用一阶保持器；

'tustin' —— 采用双线性逼近方法；

'prewarp' ——采用改进的 tustin 方法；

'matched' ——采用 SISO 系统的零极点匹配法；缺省时，method='zof '。

【例 7-11】接【例 7-4】题，试对系统采用零阶保持器与双线性逼近变换法求其离散化状态方程与离散化输出方程，设采样周期 T=1s。

【解】

为了求解离散化状态方程与离散化输出方程，特给出利用函数 c2d（）的程序 L7211.m。

```
% MATLAB PROGRAM L7211.m
%
num=[0,0,1];den=[1,3,2];s1=tf(num,den);
t=1;disp('Discrete System--using c2d with zoh')
sys1=c2d(s1,t,'zof');szoh=ss(sys1)
t=1;disp('Discrete System--using c2d with tustin')
sys2=c2d(s1,t,'tustin');sprew=ss(sys2)
```

程序运行后得：

```
Discrete System--using c2d with zoh
a =
                   x1          x2
          x1    0.50321   -0.099574
          x2        0.5           0
b =
                   u1
          x1      0.5
          x2        0
c =
                   x1          x2
          y1    0.39958     0.29399
d =
                   u1
          y1        0
Sampling time: 1
Discrete-time model.
Discrete System--using c2d with tustin
a =
                   x1          x2
          x1    0.33333           0
          x2       0.25           0
b =
                   u1
          x1        1
          x2        0
```

```
c =
                    x1          x2
        y1     0.19444     0.33333
d =
                    u1
        y1    0.083333
Sampling time: 1
Discrete-time model.
```

由以上计算数据得系统采用零阶保持器的离散化状态方程与离散化输出方程分别为：

$$\begin{bmatrix} x_1[(k+1)T] \\ x_2[(k+1)T] \end{bmatrix} = \begin{bmatrix} 0.50321 & 0.099574 \\ 0.5 & 0 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \boldsymbol{u}(kT)$$

与

$$y_1(kT) = [0.39958 \quad 0.29399]\boldsymbol{x}(kT)$$

由以上计算数据得系统采用双线性逼近变换法的离散化状态方程与离散化输出方程分别为：

$$\begin{bmatrix} x_1[(k+1)T] \\ x_2[(k+1)T] \end{bmatrix} = \begin{bmatrix} 0.33333 & 0 \\ 0.25 & 0 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \boldsymbol{u}(kT)$$

与

$$y_1(kT) = [0.19444 \quad 0.33333]\boldsymbol{x}(kT) + 0.083333u_1(kT)$$

【例 7-12】接已知连续系统的零极点增益模型为：

$$G(s) = \frac{10(s+5)}{(s+1)(s+3)(s+8)}$$

试采用一阶保持器与零极点匹配法求其离散化状态方程与离散化输出方程，设采样周期 T=0.1 s。

【解】

为了求解离散化状态方程与离散化输出方程，特给出利用函数 c2d（）的程序 L7212.m。

```
% MATLAB PROGRAM L7212.m
%
k=10;z=[-5];p=[-1,-3,-8];s=zpk (z,p,k) ;s0=tf (s) ;
t=0.1;
disp ('Discrete System--using c2d with foh')
s1=c2d (s0,t,'foh') ;ss (s1)
disp ('Discrete System--using c2d with matched')
s2=c2d (s0,t,'matched') ;ss (s2)
```

程序运行结果：

```
Discrete System--using c2d with foh
a =
                 x1          x2          x3
        x1    2.095    -0.70488     0.30119
        x2        2           0           0
        x3        0         0.5           0
b =
                 u1
        x1     0.25
        x2        0
        x3        0
c =
                 x1          x2          x3
        y1   0.2754   -0.077916   -0.007152
d =
                 u1
        y1 0.014113
Sampling time: 0.1
Discrete-time model.
Discrete System--using c2d with matched
a =
                 x1          x2          x3
        x1    2.095    -0.70488     0.30119
        x2        2           0           0
        x3        0         0.5           0
b =
                 u1
        x1     0.25
        x2        0
        x3        0
c =
                 x1          x2          x3
        y1  0.14383    0.028296   -0.087236
d =
                 u1
        y1        0
Sampling time: 0.1
Discrete-time model.
```

由以上计算数据可以得到采用不同变换方法时的系统离散化状态方程。

若在变换中输入端采用一阶保持器，其系统离散化状态方程与离散化输出方程分别为：

$$\begin{bmatrix} x_1[(k+1)T] \\ x_2[(k+1)T] \\ x_3[(k+1)T] \end{bmatrix} = \begin{bmatrix} 2.095 & 0.70488 & 0.30119 \\ 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \\ x_3(kT) \end{bmatrix} + \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{u}(kT)$$

及

$$y_1(kT) = [0.2753 \quad -0.077916 \quad -0.007152]\boldsymbol{x}(kT) + 0.014113\,\boldsymbol{u}_1(kT)$$

若在变换中输入端采用零极点匹配法，其系统离散化状态方程与离散化输出方程分别为：

$$\begin{bmatrix} x_1[(k+1)T] \\ x_2[(k+1)T] \\ x_3[(k+1)T] \end{bmatrix} = \begin{bmatrix} 2.095 & 0.70488 & 0.30119 \\ 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \\ x_3(kT) \end{bmatrix} + \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{u}(kT)$$

及

$$\boldsymbol{y}_1(kT) = [0.14383 \quad 0.028296 \quad -0.087236]\boldsymbol{x}(kT)$$

7.3 系统状态方程的线性变换

7.3.1 传递矩阵

设系统的状态方程为：

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} \tag{7-31}$$

系统的输出方程为：

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du} \tag{7-32}$$

$\boldsymbol{x}$ 为 n 维状态向量，$\boldsymbol{u}$ 为 r 维输入向量，$\boldsymbol{y}$ 为 m 维输出向量，$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 分别为 $n\times n$、$n\times r$、$m\times n$、$m\times r$ 常数矩阵。

对输入方程式（7-31）两边进行拉氏变换，则有：

$$s\boldsymbol{x}(s) - \boldsymbol{x}(0) = \boldsymbol{Ax}(s) + \boldsymbol{Bu}(s)$$

移项并整理有：

$$[s\boldsymbol{I} - \boldsymbol{A}]\boldsymbol{x}(s) = \boldsymbol{x}(0) + \boldsymbol{Bu}(s)$$

以 $[s\boldsymbol{I} - \boldsymbol{A}]^{-1}$ 式左乘上式等号两边，可得：

$$\boldsymbol{x}(s) = [s\boldsymbol{I} - \boldsymbol{A}]^{-1}\boldsymbol{x}(0) + [s\boldsymbol{I} - \boldsymbol{A}]^{-1}\boldsymbol{Bu}(s) \tag{7-33}$$

再对于输出方程式（7-32）两边进行拉氏变换，则有：

$$\boldsymbol{y}(s) = \boldsymbol{Cx}(s) + \boldsymbol{Du}(s) \tag{7-34}$$

将式（7-33）代入式（7-34）则有：

$$y(s)=C[sI-A]^{-1}x(0)+\{C[sI-A]^{-1}B+D\}u(s) \tag{7-35}$$

若初始状态为零，式（7-35）则为：

$$y(s)=\{C[sI-A]^{-1}B+D\}u(s) \tag{7-36}$$

由式（7-36）可定义传递矩阵为：

$$G(s)\overset{\text{def}}{=}\frac{y(s)}{u(s)}=C[sI-A]^{-1}B+D \tag{7-37}$$

传递矩阵 $G(s)$ 是一个 $m\times r$ 矩阵，可用来表示系统输出与输入之间的关系，即

$$y(s)=G(s)u(s) \tag{7-38}$$

在已知线性定常系统动态方程中的 A、B、C、D 之后，可以通过式（7-37）计算传递矩阵 $G(s)$。

【例 7-13】已知系统的动态方程为：

$$\begin{cases}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & -2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}\\ \begin{bmatrix}y_1\\ y_2\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}\end{cases}$$

试求该系统的传递矩阵 $G(s)$。

【解】

已知系统的参数矩阵

$$A=\begin{bmatrix}0 & 1\\ 0 & -2\end{bmatrix},B=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},C=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},D=0$$

故有特征矩阵的逆：

$$[sI-A]^{-1}=\begin{bmatrix}s & -1\\ 0 & s+2\end{bmatrix}^{-1}=\begin{bmatrix}\dfrac{1}{s} & \dfrac{1}{s(s+2)}\\ 0 & \dfrac{1}{s+2}\end{bmatrix}$$

根据传递矩阵的公式（7-37），则有：

$$G(s)=C[sI-A]^{-1}B=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\dfrac{1}{s} & \dfrac{1}{s(s+2)}\\ 0 & \dfrac{1}{s+2}\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}=\begin{bmatrix}\dfrac{1}{s} & \dfrac{1}{s(s+2)}\\ 0 & \dfrac{1}{s+2}\end{bmatrix}$$

7.3.2 状态方程的线性变换

对于一个给定的动态系统，可以选取多组不同的状态变量，所以同一系统可以有许

多状态空间表达式，但这些状态空间表达式却描述着同一个物理现象。状态变量的不同选取，实质上是状态变量间的一种线性变换或称坐标变换。如果已知状态变量之间的变换关系，就很容易建立起系数矩阵之间的转换关系。

如果系统在 x 坐标系里的动态方程为：

$$\begin{cases}\dot{x}=Ax+Bu\\ y=Cx\end{cases} \tag{7-39}$$

系统从坐标系 x 变换到 $\bar{x}$ 坐标系时（ $x=P\bar{x}$ ），若 $\bar{x}$ 坐标系里的动态方程为：

$$\begin{cases}\dot{\bar{x}}=\bar{A}\bar{x}+\bar{B}u\\ \bar{y}=\bar{C}\bar{x}\end{cases} \tag{7-40}$$

那么两坐标系相应矩阵之间有下列关系成立

$$\begin{cases}\bar{A}=P^{-1}AP\\ \bar{B}=P^{-1}B\\ \bar{C}=CP\end{cases} \tag{7-41}$$

则称为对系统进行 P 变换。

反之，从 $\bar{x}$ 坐标系逆向变换到 x 坐标系，所进行的变换为 $\bar{x}=P^{-1}x$ ，两坐标系相应矩阵之间有以下关系

$$\begin{cases}A=P\bar{A}P^{-1}\\ B=P\bar{B}\\ C=\bar{C}P^{-1}\end{cases} \tag{7-42}$$

存在，此则称为对系统进行 P^{-1} 变换。

经线性变换之后，虽然系统的状态变量变了，状态方程也变了，但是矩阵 A 与矩阵 $\bar{A}$ 都具有相同的特征值。因为：

$$\left|sI-\bar{A}\right|=\left|sI-P^{-1}AP\right|=\left|sP^{-1}P-P^{-1}AP\right|=\left|P^{-1}sP-P^{-1}AP\right|=$$

$$P^{-1}(sI-A)P\big|=\left|P^{-1}\right|\left|sI-A\right|\left|P\right|=\left|P^{-1}P\right|\left|sI-A\right|=\left|sI-A\right|$$

这就是系统特征值的不变性。

【例 7-14】已知系统状态空间表达式为：

$$\begin{cases}\dot{x}=\begin{bmatrix}0 & -2\\ 1 & -3\end{bmatrix}x+\begin{bmatrix}2\\ 0\end{bmatrix}u\\ y=\begin{bmatrix}0 & 3\end{bmatrix}x\end{cases}$$

且 $x_1(0)=1, x_2(0)=1$ 。欲对系统进行坐标变换，其变换关系为 $P=\begin{bmatrix}6 & 2\\ 2 & 0\end{bmatrix}$，试求系统线性变换后的系统动态方程。

【解】

系统从坐标系 x 变换到 $\bar{x}$ 坐标系时（$\boldsymbol{x}=\boldsymbol{P}\bar{\boldsymbol{x}}$），其变换关系为式（7-41），为求系统的线性变换矩阵，给出程序 L7302.m 如下。

```
% MATLAB PROGRAM L7302.m
%
P=[6 2;2 0];P1=inv(P);
A=[0 -2;1 -3];A1=P1*A*P
B=[2 0]';B1=P1*B
C=[0 3];C1=C*P
```

程序运行后得 $\bar{x}$ 坐标系的系数矩阵：

```
A1 =
     0     1
    -2    -3
B1 =
     0
     1
C1 =
     6   0
```

由以上计算数据，$A1=\bar{A},B1=\bar{B},C1=\bar{C}$，（以下举例同此）于是可写出线性变换后的系统动态方程：

$$\begin{cases}\dot{\bar{x}}=\begin{bmatrix}0 & 1\\ -2 & -3\end{bmatrix}\bar{x}+\begin{bmatrix}0\\1\end{bmatrix}u\\ \bar{y}=\begin{bmatrix}6 & 0\end{bmatrix}\bar{x}\end{cases}$$

【例 7-15】对【例 7-7】之系统进行坐标变换，其变换关系为：

$$\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1 & 1 & 1\\ -1 & -2 & -3\\ 1 & 4 & 9\end{bmatrix}\begin{bmatrix}\bar{x}_1\\ \bar{x}_2\\ \bar{x}_3\end{bmatrix}$$

并验证系统特征值的不变性。

【解】

（1）对系统进行坐标变换

已知变换 $\boldsymbol{P}=\begin{bmatrix}1 & 1 & 1\\ -1 & -2 & -3\\ 1 & 4 & 9\end{bmatrix}$，系统从坐标系 x 变换到 $\bar{x}$ 坐标系时（$\boldsymbol{x}=\boldsymbol{P}\bar{\boldsymbol{x}}$），

其变换关系为式（7-41），为求系统的线性变换矩阵，给出程序 L7303.m 如下。

```
% MATLAB PROGRAM L7303.m
%
```

```
A=[0 1 0;0 0 1;-6 -11 -6];
B=[0 0 1]';
P=[1 1 1;-1 -2 -3;1 4 9];P1=inv(P)
A1=P1*A*P
B1=P1*B
```

程序运行后得 $\bar{x}$ 坐标系的系数阵：

```
P1 =
    3.0000    2.5000    0.5000
   -3.0000   -4.0000   -1.0000
    1.0000    1.5000    0.5000
A1 =
   -1.0000    0.0000   -0.0000
    0.0000   -2.0000    0.0000
   -0.0000   -0.0000   -3.0000
B1 =
    0.5000
   -1.0000
    0.5000
```

由以上计算数据可得系统 $\bar{x}$ 坐标系里的动态方程为：

$$\begin{bmatrix}\dot{\bar{x}}_1\\ \dot{\bar{x}}_2\\ \dot{\bar{x}}_3\end{bmatrix}=\begin{bmatrix}-1 & 0 & 0\\ 0 & -2 & 0\\ 0 & 0 & -3\end{bmatrix}\begin{bmatrix}\bar{x}_1\\ \bar{x}_2\\ \bar{x}_3\end{bmatrix}+\begin{bmatrix}0.5\\ -1\\ 0.5\end{bmatrix}\boldsymbol{u}$$

（2）验证系统特征值的不变性

原系统特征式为：

$$|s\boldsymbol{I}-\boldsymbol{A}|=s^3+6s^2+11s+6=(s+1)(s+2)(s+3)$$

坐标变换后系统的特征式为：

$$|s\boldsymbol{I}-\overline{\boldsymbol{A}}|=\begin{bmatrix}s & 0 & 0\\ 0 & s & 0\\ 0 & 0 & s\end{bmatrix}-\begin{bmatrix}-1 & 0 & 0\\ 0 & -2 & 0\\ 0 & 0 & -3\end{bmatrix}=(s+1)(s+2)(s+3)$$

即说明系统特征值没有改变。

【例 7-16】对【例 7-15】$\bar{x}$ 坐标系里的系统进行坐标反变换（从 $\bar{x}$ 坐标系逆向变换到 $\boldsymbol{x}$ 坐标系）。

【解】

从 $\bar{x}$ 坐标系逆向变换到 $\boldsymbol{x}$ 坐标系，需进行 $\bar{x}=\boldsymbol{P}^{-1}\boldsymbol{x}$ 变换，有关系式（7-42）：

$$\begin{cases}\boldsymbol{A}=\boldsymbol{P}\overline{\boldsymbol{A}}\boldsymbol{P}^{-1}\\ \boldsymbol{B}=\boldsymbol{P}\overline{\boldsymbol{B}}\end{cases}$$

已知 $\boldsymbol{P}1=(\text{inv}(\boldsymbol{p}))$,则 $\boldsymbol{P}1$ 与系统在 $\bar{x}$ 坐标系里的动态方程分别为：

$$\boldsymbol{P}1=\begin{bmatrix}3 & 2.5 & 0.5\\ -3 & -4 & -1\\ 1 & 1.5 & 0.5\end{bmatrix},\quad \begin{bmatrix}\dot{\bar{x}}_1\\ \dot{\bar{x}}_2\\ \dot{\bar{x}}_3\end{bmatrix}=\begin{bmatrix}-1 & 0 & 0\\ 0 & -2 & 0\\ 0 & 0 & -3\end{bmatrix}\begin{bmatrix}\bar{x}_1\\ \bar{x}_2\\ \bar{x}_3\end{bmatrix}+\begin{bmatrix}0.5\\ -1\\ 0.5\end{bmatrix}\boldsymbol{u}$$

根据式（7-42）为求系统的线性变换矩阵特给出程序 L7304.m 。

```
% MATLAB PROGRAM L7304.m
%
A1=[-1 0 0;0 -2 0;0 0 -3];B1=[0.5 -1 0.5]';
p1=[3 2.5 0.5;-3 -4 -1;1 1.5 0.5];
P=inv(P1);
A = P * A1* P1
B = P * B1
```

程序运行后可得：

```
A =
    0.0000    1.0000    0.0000
         0         0    1.0000
   -6.0000  -11.0000   -6.0000
B =
   -0.0000
    0.0000
    1.0000
```

将矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$ 代入式（7-39），则有：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -6 & -11 & -6\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}\boldsymbol{u}$$

上式即为 $\boldsymbol{x}$ 坐标系里的状态方程。

【例 7-17】已知系统的状态矩阵为：

$$\boldsymbol{A}=\begin{bmatrix}2 & 1 & -1\\ 1 & 2 & -1\\ -1 & -1 & 2\end{bmatrix}$$

试将矩阵 $\boldsymbol{A}$ 化为对角矩阵。

【解】

可以先求矩阵 $\boldsymbol{A}$ 的特征向量与特征值，再求特征向量矩阵 $\boldsymbol{V}$ 的逆，由特征向量矩阵 $\boldsymbol{V}$ 组成的变换矩阵就能将矩阵 $\boldsymbol{A}$ 化为对角矩阵。按照这个思路给出如下程序 L7305.m。

```
% MATLAB PROGRAM L7305.m
%
A=[2 1 -1;1 2 -1;-1 -1 2];
[V,D]=eig(A);
V1 = inv(V);
Ad=V1*A*V
```

运行程序后可将矩阵 $\boldsymbol{A}$ 化为对角矩阵：

```
Ad =
    1.0000   -0.0000   -0.0000
   -0.0000    1.0000   -0.0000
         0   -0.0000    4.0000
```

7.3.3 范德蒙特矩阵与约当标准形

如果状态矩阵 $\boldsymbol{A}$ 具有如下的标准形式

$$\boldsymbol{A}=\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \tag{7-43}$$

并且 $\boldsymbol{A}$ 又具有各异的特征值 λ_1、λ_2、……、λ_n，则以下范德蒙特矩阵 $\boldsymbol{P}$

$$\boldsymbol{P}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \lambda_3 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \lambda_3^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix} \tag{7-44}$$

可使 $\boldsymbol{A}$ 对角化。

如果矩阵 $\boldsymbol{A}_{n\times n}$ 有相重的特征值，并且按 λ_i 所解得的独立特征向量的数目小于 n，则矩阵 $\boldsymbol{A}$ 不能化为对角阵。但仍存在一个线性变换矩阵 $\boldsymbol{P}$，使 $\boldsymbol{A}$ 变换成

$$\boldsymbol{J}=\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P} \tag{7-45}$$

$\boldsymbol{J}$ 矩阵接近是一个对角矩阵，叫做约当（Jordan）标准形矩阵，简称约当标准形。例如：

$$\boldsymbol{J}=\left[\begin{array}{ccc|cc|c} \lambda_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & \lambda_2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \lambda_2 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & \lambda_3 \end{array}\right] \tag{7-46}$$

设λ_j为K重根，对应于λ_j的约当块为：

$$\begin{bmatrix} \lambda_j & 1 & & 0 \\ & \lambda_j & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_j \end{bmatrix}_{K\times K} \tag{7-47}$$

而与λ_j相对应的独立特征向量为：

$$\boldsymbol{p}_j = \begin{bmatrix} 1 & \lambda_j & \lambda_j^2 & \cdots & \lambda_j^{n-1} \end{bmatrix}^T \tag{7-48}$$

则与约当块相对应的变换矩阵部分为：

$$\boldsymbol{P} = \begin{bmatrix} \cdots & \boldsymbol{P}_j & \dfrac{\mathrm{d}\boldsymbol{P}_j}{\mathrm{d}\lambda_j} & \dfrac{\mathrm{d}^2\boldsymbol{P}_j}{\mathrm{d}\lambda_j^2} & \cdots & \dfrac{\mathrm{d}^{k-1}\boldsymbol{P}_j}{\mathrm{d}\lambda_j^{k-1}} & \cdots \end{bmatrix} \tag{7-49}$$

【例 7-18】已知控制系统为：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & -5 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

试将矩阵$\boldsymbol{A}$化为约当标准形（矩阵）。

【解】

已知矩阵$\boldsymbol{A}$为式（7-43）标准形，可使用以下指令求矩阵$\boldsymbol{A}$的特征向量与特征值：

```
A=[0 1 0;0 0 1 ;2 -5 4]
[V,D]=eig (A)
V =
    0.5774   -0.5774    0.2182
    0.5774   -0.5774    0.4364
    0.5774   -0.5774    0.8729
D =
    1.0000         0         0
         0    1.0000         0
         0         0    2.0000
```

由运算数据可得$\lambda_1=\lambda_2=1, \lambda_3=2$。利用公式（7-48）与（7-49）有：

$$\boldsymbol{p}_1 = \begin{bmatrix} 1 \; \lambda_1 \; \lambda_1^2 \end{bmatrix}^T, \quad \boldsymbol{p}_3 = \begin{bmatrix} 1 \; \lambda_3 \; \lambda_3^2 \end{bmatrix}^T$$

$$\boldsymbol{P} = \begin{bmatrix} \cdots & \boldsymbol{p}_j & \dfrac{dp_j}{d\lambda_j} & \dfrac{d^2p_j}{d\lambda_j^2} & \cdots & \dfrac{d^{K-1}p_j}{d\lambda_j^{K-1}} & \cdots \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 1 \\ \lambda_1 & 1 & \lambda_3 \\ \lambda_1^2 & 2\lambda_1 & \lambda_3^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 4 \end{bmatrix}$$

利用公式（7-45），给出求矩阵$\boldsymbol{A}$变换矩阵的指令：

```
A=[0 1 0;0 0 1 ;2 -5 4];
P=[1 0 1;1 1 2;1 2 4];
P1=inv (P) ;
J=P1*A*P
```

指令执行后可得约当阵：

```
J =
    1 1 0
    0 1 0
    0 0 2
```

计算的约当阵是否正确，可用 MATLAB 的约当函数 jordan（）进行校验：

```
A=[0 1 0;0 0 1 ;2 -5 4];
J=jordan (A)
```

运算表明计算的约当标准形正确。

```
J =
    2 0 0
    0 1 1
    0 0 1
```

【例 7-19】已知控制系统与变换矩阵分别为:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 5 & 4 & 0 \\ 0 & 1 & 0 \\ -4 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} 0 & 4 & 16 \\ 0 & -4 & 0 \\ -32 & 4 & -16 \end{bmatrix}$$

试将系统变换成约当标准形。

【解】

已知系统状态矩阵 $\boldsymbol{A}$ 与变换矩阵 $\boldsymbol{P}$，将系统变换成约当标准形可执行以下指令：

```
A=[5 4 0;0 1 0;-4 4 1];
P=[0 4 16;0 -4 0;-32 4 -16];
P1=inv (P) ;J=P1*A*P
J =
    1 1 0
    0 1 0
    0 0 5
```

检验系统转换为约当标准形是否正确，可执行以下指令：

```
A=[5 4 0;0 1 0;-4 4 1]; J=jordan (A)
J =
    5 0 0
    0 1 1
    0 0 1
```

指令执行结果说明转换正确。

7.4 线性系统的可控性与可观性

在状态空间分析中，对系统的描述可以状态方程与输出方程来表示。这种表示法的突出特点是能够描述系统内部状态的变化。基于状态空间分析的系统可控性与可观性也是非常重要的概念。

所谓系统的可控性是指系统的状态是可以被控制还是不能被控制；而系统的可观（测）性则是指系统状态的变化能够由输出检测反映出来还是不能由输出检测反映出来。

系统的可控性与可观性对系统的设计至关重要。在设计一个最优控制系统之前，必须首先知道系统是否可控以及是否可观测，这是因为系统的可控性与可观测性往往是确定最优系统是否有解的先决条件。

7.4.1 线性系统的可控性

一个线性定常系统（$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$），其状态向量 $\boldsymbol{x}$ 属于 n 维实空间，即 $\boldsymbol{x}\in \boldsymbol{R}^n$。若对 $\boldsymbol{R}^n$ 空间中任一状态 $\boldsymbol{x}(t_0)$ 与 $\boldsymbol{R}^n$ 空间中另一任意状态 $\boldsymbol{x}(t_f)$，存在一个有限的时间 $t_f > t_0$ 及输入 $\boldsymbol{u}(t_0,t_f)$，能在 t_f 内使 $\boldsymbol{x}(t_0)$ 状态转移到 $\boldsymbol{x}(t_f)$，则系统叫做完全可控或简称可控。

可控性的概念仅要求输入 $\boldsymbol{u}$ 能在有限的时间内，使系统由状态空间的任意一状态转移到另一个任意状态。没有规定转移的路线轨迹，也没有限制输入量的大小。

如果一个系统至少有一个状态变量是不可控的，则系统叫做不完全可控，或简称不可控。

7.4.1.1 离散系统的完全可控性

离散线性定常系统：

$$\begin{cases} \boldsymbol{x}(k+1)=\boldsymbol{F}\boldsymbol{x}(k)+\boldsymbol{G}\boldsymbol{u}(k) \\ \boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)+\boldsymbol{D}\boldsymbol{u}(k) \end{cases} \tag{7-50}$$

式中 $\boldsymbol{x}(k)$ 为 n 维状态向量，$\boldsymbol{u}(k)$ 为 r 维输入量，$\boldsymbol{F}$ 为 $n\times n$ 非奇异阵；$\boldsymbol{G}$ 为 $n\times r$ 矩阵（若为单输入时 r=1），状态向量系统完全可控的必要且充分的条件是下列可控性矩阵

$$\boldsymbol{CO}=[\boldsymbol{G} \mid \boldsymbol{FG} \mid \cdots \mid \boldsymbol{F}^{n-1}\boldsymbol{G}]$$

之秩为 n，即

$$rank[\boldsymbol{G} \mid \boldsymbol{FG} \mid \cdots \mid \boldsymbol{F}^{n-1}\boldsymbol{G}]=n \tag{7-51}$$

可以用 MATLAB 求可控性矩阵的函数 ctrb（ ）求 $n\times nr$（nr 意为 $n\times r$ 之值，下同）可控性矩阵 $\boldsymbol{CO}$。函数 ctrb （ ）的调用格式为：

CO= ctrb（A,B）

其中，输入参量 $\boldsymbol{A}$ 为离散系统的系统矩阵 $\boldsymbol{F}$ 或者连续系统的系统矩阵 $\boldsymbol{A}$，输入参量 $\boldsymbol{B}$ 为离散系统的控制矩阵 $\boldsymbol{G}$ 或者连续系统的控制矩阵 $\boldsymbol{B}$，函数返回的就是系统可控性矩阵 $\boldsymbol{CO}$。可见函数 ctrb （ ） 既适用于离散系统，也适用于连续系统。

【例 7-20】已知离散系统方程为：

$$\begin{cases} x(k+1) = fx(k) + gu(k) \\ y(k) = cx(k) + du(k) \end{cases}$$

已知方程式中：**f**=[0.9048 0 0；0.1338 0.4651 -0.2237；0.0243 0.2237 0.9602]；**g**=[0.0952；0.0784；0.0135]；**c**=[0 0 2.5298]；**d**=0；设采样周期 T=0.1 s，试确定离散系统的可控性。

【解】

已知离散系统方程的常数矩阵（**a**、**b**、**c** 与 **d**），计算系统的可控性矩阵，最后计算可控性矩阵的秩。根据状态完全可控必要且充分的条件即可控性矩阵的秩为系统的阶次 n，来确定系统的可控性。由这些分析给出以下程序 L7401.m。

```
% MATLAB PROGRAM L7401.m
%
f=[0.9048 0 0;0.1338 0.4651 -0.2237;0.0243 0.2237 0.9602];a=f;
g=[0.0952;0.0784;0.0135];b=g;
c=[0 0 2.5298];
d=0;n=3;
CAM=ctrb (a,b) ;
rcam=rank (CAM) ;
if rcam ==n
disp ('System is controlled')
elseif rcam<n
disp ('System is no controlled')
end
```

程序运行后其结论是系统完全可控：

```
System is controlled
```

7.4.1.2 连续系统的完全可控性

设线性系统的动态方程为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

式中 A、B、C、D 分别为 $n \times n$、 $n \times r$ 、 $m \times n$ 与 $m \times r$ 常数矩阵。

系统完全可控的条件是下列 $n \times nr$ 可控性矩阵

$$CO = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}$$

的秩为 n，即

$$rank\begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} = n \tag{7-52}$$

【例 7-21】已知三维状态方程

$$\dot{x}=\begin{bmatrix}1&1&0\\0&1&0\\0&1&1\end{bmatrix}x+\begin{bmatrix}0&1\\1&0\\0&1\end{bmatrix}u$$

试确定系统的可控性。

【解】

已知系统矩阵 A 与 B，求系统的能控矩阵可执行以下指令：

```
A=[1 1 0;0 1 0;0 1 1];B=[0 1;1 0; 0 1]
CAM=ctrb (A,B)
CAM =
     0     1     1     1     2     1
     1     0     1     0     1     0
     0     1     1     1     2     1
```

判断系统的可控性需执行以下指令求可控性矩阵的秩：

```
rcam=rank (CAM)
rcam =
 2
```

根据公式（7-52）有：

$$\text{rank}\begin{bmatrix}B & AB & A^2B & \cdots & A^{n-1}B\end{bmatrix}=2<n(=3)$$

所以该系统是不可控的。解算本题也可执行以下程序 L7402.m。

```
% MATLAB PROGRAM L7402.m
%
A=[1 1 0;0 1 0;0 1 1];
B=[0 1;1 0; 0 1]; n=3;
CAM=ctrb (A,B) ;
rcam=rank (CAM) ;
if rcam ==n
    disp ('System is controlled')
elseif rcam<n
    disp ('System is no controlled')
end
```

7.4.1.3 连续系统的状态完全可控标准形

对于单输入系统 $\{A,b\}$，若其状态矩阵与输入矩阵具有如下的标准形式：

$$A=\begin{bmatrix}0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\vdots&&\vdots\\0&0&0&\cdots&1\\-a_0&-a_1&-a_2&\cdots&-a_{n-1}\end{bmatrix},\quad b=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}\tag{7-53}$$

则系统一定可控，该矩阵对 $(\boldsymbol{A},\boldsymbol{b})$ 被叫做可控标准形。

若 n 维单输入线性系统可控，则一定能找到一个线性变换将系统变换成可控标准形。其变换矩阵 $\boldsymbol{P}$ 可按以下步骤求得。

系统可控性矩阵

$$\boldsymbol{S}=\begin{bmatrix}\boldsymbol{b} & \boldsymbol{A}\boldsymbol{b} & \boldsymbol{A}^2\boldsymbol{b} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b}\end{bmatrix}$$

为非奇异，其逆矩阵一定存在。若设

$$\boldsymbol{S}^{-1}=c\cdot\begin{bmatrix}s_1^T\\ s_2^T\\ \vdots\\ s_n^T\end{bmatrix} \tag{7-54}$$

式中 $\boldsymbol{s}_1$、$\boldsymbol{s}_2$、……、$\boldsymbol{s}_n$ 为 $n\times1$ 列向量；c 为任意常数，则用 $\boldsymbol{s}_n^T$、$\boldsymbol{s}_n^T\boldsymbol{A}$、……、$\boldsymbol{s}_n^T\boldsymbol{A}^{n-1}$ 构成的下述矩阵即为变换矩阵

$$\boldsymbol{P}=\begin{bmatrix}\boldsymbol{s}_n^T\\ \boldsymbol{s}_n^T\boldsymbol{A}\\ \vdots\\ \boldsymbol{s}_n^T\boldsymbol{A}^{n-1}\end{bmatrix} \tag{7-55}$$

【例 7-22】已知系统状态方程为：

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}u$$

式中
$$\boldsymbol{A}=\begin{bmatrix}-2 & 2 & -1\\ 0 & -2 & 0\\ 1 & -4 & 0\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}0\\1\\1\end{bmatrix}$$

试将系统状态方程化为可控标准形。

【解】

（1）将系统状态方程化为可控标准形

根据已知系统参数，首先计算系统的可控形矩阵及其行列式的值。若可控形矩阵的行列式的值≠0，必可计算系统可控形矩阵的逆；根据公式（7-55），系统可控形矩阵的逆的最后一行是变换矩阵的第一行；按公式（7-55）依次计算变换矩阵的第二行第三行；再根据以上数据可求出变换矩阵及其逆矩阵；最后计算系统可控标准形的矩阵对。根据这个思路给出以下 MATLAB 程序 L7403a.m。

```
% MATLAB PROGRAM L7403a.m
%
A=[-2 2 -1;0 -2 0;1 -4 0];b=[0 1 1]';S=ctrb(A,b);
if det(S)~=0
   s1=inv(S);
end
```

```
P=[s1(3,:);s1(3,:)*A;s1(3,:)*A*A]
P1=inv(P)
A1=P*A*P1
B1=P*b
```

执行以上程序可得：

```
A1 =
     0     1     0
     0     0     1
    -2    -5    -4
b1 =
     0
     0
     1
```

则系统的可控标准形的矩阵对为：

$$\overline{A}=A1=\begin{bmatrix}0&1&0\\0&0&1\\-2&-5&-4\end{bmatrix},\quad \overline{B}=b1=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

（2）验证变换后的系统是否可控

为验证变换后的系统是否可控，给出以下程序 L7403b.m：

```
% MATLAB PROGRAM L7403b.m
%
A=[-2 2 -1;0 -2 0;1 -4 0];b=[0 1 1]';S=ctrb(A,b);
if det(S)~=0
   s1=inv(S);
if rcam ==n
   disp('System is controlled')
elseif rcam<n
   disp('System is no controlled')
end
```

程序运行后的结论是系统完全可控：

```
System is controlled
```

7.4.1.4 连续系统的输出可控性

控制系统的设计中，有时需要控制系统的输出，而不是控制系统的状态。对于输出控制而言，状态完全可控既不是必要的，也不是充分的。因此需要定义系统的输出完全可控性。

若系统的状态方程为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

式中 A 、 B 、 C 与 D 分别为 $n\times n$ 、 $n\times r$ 、 $m\times n$ 与 $m\times r$ 常数矩阵。

如果存在一个无约束的分段连续的控制向量 $u(t)$，在有限时间 $(t_f - t_0)$ 内，使得任一输出 $y(t_0)$ 能够转移到 $y(t_f)$，则称这个系统为输出完全可控。

上述系统输出完全可控的条件为下列 $m\times(n+1)r$ 矩阵:

$$T = [CB \mid CAB \mid CA^2B \mid \cdots \mid CA^{n-1}B \mid D]$$

的秩为 m，即

$$\text{rank}(T) = \text{rank}[CB \mid CAB \mid CA^2B \mid \cdots \mid CA^{n-1}B \mid D] = m \qquad (7\text{-}56)$$

T 称为输出可控性矩阵。

【例 7-23】已知系统动态方程为：

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \\ y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{cases}$$

试确定系统状态的可控性与输出可控性。

【解】

（1）确定系统状态的可控性

给出以下程序 L7404a.m 即可确定系统状态的可控性。

```
% MATLAB PROGRAM L7404a.m
%
A=[0 1;-1 -2];b=[1;-1];C=[1 0];d=0;n=2;
CAM=ctrb(A,b);
if det(CAM)~=0
   rcam=rank(CAM)
   if rcam ==n
        disp('System is controlled')
   elseif rcam<n
        disp('System is no controlled')
end
elseif det(CAM)==0
   disp('System is no controlled')
end
```

程序运行的结论是系统状态不可控：

```
System is no controlled
```

（2）确定系统输出可控性

给出以下程序 L7404b.m 即可确定系统输出的可控性。

```
% MATLAB PROGRAM L7404b.m
%
A=[0 1;-1 -2];b=[1;-1];c=[1 0];d=0;
n=2;p=1;m=1;t1=c*b;t2=c*A*b;
T=[t1 t2 d]
rcam=rank (T)
if rcam ==m
    disp ('System Output is controlled')
elseif rcam~=m
    disp ('System Output is no controlled')
end
```

程序运行的结论是输出可控：

```
System Output is controlled
```

7.4.2 线性系统的可观测性

若 n 维线性定常系统的动态方程为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

式中 A 、 B 、 C 、 D 分别为 $n \times n$ 、 $n \times r$ 、 $m \times n$ 与 $m \times r$ 常数矩阵。

如果在有限时间间隔 $t_0 \leqslant t \leqslant t_f$ 内，根据输出值 $y(t)$ 与给出的 $u(t)$，能够确定系统的初始状态 $x(t_0)$ 的每一个分量，则叫此系统为完全可观测的，简称可观测的。若系统中至少有一个状态变量是不可观测的，则称此系统为不完全可观测，或简称不可观测。

7.4.2.1 离散系统的完全可观测性

设 n 维离散系统：

$$\begin{cases} x(k+1) = Fx(k) + Gu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

式中 F 、 G 、 C 、 D 为 $n \times n$ 、 $n \times r$ 、 $m \times n$ 与 $m \times r$ 矩阵。系统完全可观测的条件为以下 $nm \times n$ 矩阵 $OB1$ 的秩等于 n。即

$$OB1 = \begin{bmatrix} C \\ CF \\ \vdots \\ CF^{n-1} \end{bmatrix}, \quad \text{rank}(OB1) = n \tag{7-57}$$

可以用 MATLAB 的求可观性矩阵函数 obsv （ ）来计算 $nm \times n$ 矩阵 OB（nm 意为 n×m 之值，下同）。函数 obsv （ ） 的调用格式为：

$$OB = \text{obsv}（A,C）$$

其中，输入参量 A 即为离散系统的系统矩阵 F 或者连续系统的系统矩阵 A，输入参量 C 即为离散系统的观测矩阵或者连续系统的观测矩阵，函数返回的就是系统可观性矩阵 OB。可见函数 obsv（ ）既适用于离散系统，也适用于连续系统。

由于矩阵与其转置矩阵的秩是相同的，故系统完全可观测的条件又可为以下 $n\times nm$ 矩阵 $OB2$ 的秩等于 n。即

$$OB2=[C^T \quad F^TC^T \quad \cdots \quad (F^T)^{n-1}C^T] \text{与} \operatorname{rank}(OB2)=n \tag{7-58}$$

【例 7-24】接【例 7-20】，已知三维离散系统状态方程，试确定系统的可观测性。

【解】

为确定系统的可观测性，给出调用函数 obsv（ ）的如下程序 L7405.m。

```
% MATLAB PROGRAM L7405.m
%
a=[0.9048 0 0;0.1338 0.4651 -0.2237;0.0243 0.2237 0.9602];
b=[0.0952;0.0784;0.0135];
c=[0 0 2.5298];d=0;n=3;
ob=obsv(a,c);
roam=rank(ob);
  if roam ==n
    disp('System is observable')
  elseif roam~=n
    disp('System is no observable')
end
```

程序运行的结论是系统是可观的：

```
System is observable
```

7.4.2.2 连续系统的完全可观测性

设连续系统的动态方程为：

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

式中 A、B、C、D 分别为 $n\times n$、$n\times r$、$m\times n$ 与 $m\times r$ 常数矩阵。系统完全可观测的条件为以下 $nm\times n$ 矩阵 $OB1$ 的秩等于 n。即

$$OB1=\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}, \quad \operatorname{rank}(OB1)=n \tag{7-59}$$

或者以下 $n\times nm$ 可观测矩阵 $OB2$ 的秩等于 n。即

$$OB2=\begin{bmatrix} C^T & A^TC^T & \cdots & (A^T)^{n-1}C^T \end{bmatrix}, \quad \operatorname{rank}(OB2)=n \tag{7-60}$$

【例 7-25】 已知线性系统动态方程为：

$$\begin{cases} \dot{x} = \begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix} x + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} x \end{cases}$$

试判别系统的可控性及可观测性。

【解】

为判别系统的可控性及可观测性，给出利用函数 ctrb （ ）、obsv （ ） 的如下程序 L7406.m。

```
% MATLAB PROGRAM L7406.m
%
A=[-3 1;1 -3];B=[1 1;1 1];C=[1 1;1 -1];D=[0];
CAM=ctrb(A,B);rcam=rank(CAM);n=2;
    if rcam ==n
        disp('System is controlled')
    elseif rcam<n
        disp('System is no controlled')
    end
ob=obsv(A,C);
roam=rank(ob);
    if roam ==n
        disp('System is observable')
    elseif rcam~=n
        disp('System is no observable')
    end
```

程序运行结果为：

```
System is no controlled
System is observable
```

即系统为不可控但是可观。

【例 7-26】已知系统动态方程为：

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{cases}$$

试判别系统的可控性及可观测性。

【解】

为判别系统的可控性及可观测性，给出程序 L7407.m 如下。

```
% MATLAB PROGRAM L7407.m
%
A=[-2 0;0 -1];B=[3;1];C=[1 0];D=[0];
CAM=ctrb(A,B);rcam=rank(CAM);n=2;
    if rcam ==n
       disp('System is controlled')
    elseif rcam<n
        disp('System is no controlled')
    end
ob=obsv(A,C);
roam=rank(ob);
    if roam ==n
        disp('System is observable')
    else
        disp('System is no observable')
    end
```

程序运行结果为:

```
System is controlled
System is no observable
```

即系统为可控而不可观测的。

【例 7-27】已知连续系统的传递函数模型为:

$$G(s)=\frac{s+\alpha}{s^3+10s^2+27s+18}$$

当α 分别取 -1，0，+1 时。判别系统的可控性与可观测性。

【解】

为判别系统的可控性及可观测性，给出如下程序 L7408.m。

```
% MATLAB PROGRAM L7408.m
%
for alph=[-1:1]
     alph
     num=[1,alph];
     den=[1 10 27 18];
     [A,B,C,D]=tf2ss(num,den);
     cam=ctrb(A,B); rcam=rank(cam);
     ob=obsv(A,C);roam=rank(ob);
     CAM=ctrb(A,B);rcam=rank(CAM);n=3;
if rcam ==n
   disp('System is controlled')
elseif rcam<n
```

```
    disp('System is no controlled')
end
ob=obsv(A,C);
roam=rank(ob);
if roam ==n
     disp('System is observable')
else
     disp('System is no observable')
end
end
```

程序运行结果：

```
alph = -1
System is controlled
System is observable
alph = 0
System is controlled
System is observable
alph = 1
System is controlled
System is no observable
```

即当α =-1与α =0时，系统为可控与可观测；当α =1时，系统为可控而不可观。

7.4.2.3 连续系统的完全可观测标准形

对于单输出系统$\{A,B,c,d\}$，若其状态矩阵与输出矩阵有如下的标准形式：

$$A=\begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix},\quad c=[0\,0\cdots 0\,1] \tag{7-61}$$

则该系统一定可观测，而矩阵对(A,c)被叫做可观标准形。

若单输入单输出系统可观测，则一定有一个线性变换M

$$x=M\bar{x} \tag{7-62}$$

可将系统转换为可观标准形：

$$\begin{cases} \dot{\bar{x}}=\bar{A}\bar{x}+\bar{B}u & (7\text{-}63) \\ y=\bar{c}\bar{x} & (7\text{-}64) \end{cases}$$

而以上两式中$\bar{A}$与$\bar{c}$将具有式（7-60）的标准形式。

若系统可观测，则有可观测矩阵V

$$V=\begin{bmatrix}C^T & A^TC^T & \cdots & (A^T)^{n-1}C^T\end{bmatrix} \tag{7-65}$$

其逆矩阵 V^{-1} 一定存在。设

$$V^{-1}=\begin{bmatrix}v_1^T\\ v_2^T\\ \vdots\\ v_n^T\end{bmatrix} \tag{7-66}$$

通过矩阵 P 可求得变换矩阵 M，即有：

$$P=\begin{bmatrix}v_1^T\\ v_2^TA^T\\ \vdots\\ v_n^T(A^T)^{n-1}\end{bmatrix} \tag{7-67}$$

与

$$M=P^T \tag{7-68}$$

进而求得可观标准形的矩阵对：

$$\begin{cases}\bar{A}=M^{-1}AM=P^{-T}AP^T & (7\text{-}69)\\ \bar{c}=cM=cP^T & (7\text{-}70)\end{cases}$$

【例 7-28】已知系统动态方程为：

$$\begin{cases}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}-1\\ 1\end{bmatrix}u\\ y=\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}\end{cases}$$

是将系统的动态方程化为可观测标准形，并求出其变换矩阵。

【解】

为了将系统化为可观测标准形并求其变换矩阵，根据关系式（7-65）~（7-70）给出程序 L7409.m 如下。

```
% MATLAB PROGRAM L7409.m
%
A=[1 -1;1 1];B=[-1;1];C=[1 1];
V=obsv(A,C);rank(V);
V1=inv(V);V10=(V1)';
V11=(V10(2,:));VA=(V11)*(A)';
N=[V11;VA];M=N'
A1=inv(M)*A*M
c1=C*M
```

程序运行后得：

```
M =
    0.5000    1.0000
   -0.5000         0
A1 =
     0    -2
     1     2
c1 =
     1
```

$\overline{A}=A1, \overline{c}=c1$，即有可观标准形的矩阵对：

$$\overline{A}=\begin{bmatrix}0 & -2\\1 & 2\end{bmatrix},\quad \overline{c}=\begin{bmatrix}0 & 1\end{bmatrix}$$

还有变换矩阵

$$M=\begin{bmatrix}0.5 & 1\\-0.5 & 0\end{bmatrix}$$

7.4.3 线性定常系统的实现问题

若已知系统的 A、B、C、D 4个矩阵，可由再次给出的下式求出传递矩阵

$$G(s)=C[sI-A]^{-1}B+D \tag{7-71}$$

线性系统的实现问题正好是以上命题的逆命题，即给定系统的传递矩阵，能否求出系统的状态方程与输出方程，也就是给定 G（s），求 A、B、C、D 阵的问题。

系统传递函数通常表示为有理分式，其分子的次数等于或小于分母的次数时，叫做真分式传递函数；如果分子的次数小于分母的次数时，叫做严格真分式传递函数。

对于 SISO 线性定常系统，如果其传递函数是可实现的，则有无穷多个不同的状态方程实现。其阶数最小的实现叫做最小实现，最小实现的阶数等于传递函数的次数。

任何有理真分式传递函数都可表示为：

$$G(s)=\frac{b_n s^n+b_{n-1}s^{n-1}+\cdots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}=d'+\frac{\beta_{n-1}s^{n-1}+\beta_{n-2}s^{n-2}+\cdots+\beta_1 s+\beta_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}=$$

$$d'+\frac{N(s)}{D(s)}=d'+\hat{G}(s) \tag{7-72}$$

若系统 $\{A,b,c,d\}$ 是 G（s）的一个实现，则可导出

$$d=d'$$

其可控标准形为：

$$A=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\0 & 0 & 1 & \cdots & 0\\\vdots & \vdots & \vdots & & \vdots\\0 & 0 & 0 & \cdots & 1\\-a_0 & -a_1 & -a_2 & \cdots & -a_{n-1}\end{bmatrix},\quad b=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix} \tag{7-73}$$

$$c=\left[\beta_0\ \beta_1\ \beta_2\cdots\beta_{n-1}\right] 与 d=d' \tag{7-74}$$

其可观标准形为：

$$A=\begin{bmatrix}0 & 0 & \cdots & 0 & -a_0\\ 1 & 0 & \cdots & 0 & -a_1\\ 0 & 1 & \cdots & 0 & -a_2\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \cdots & 1 & -a_{n-1}\end{bmatrix},\quad b=\begin{bmatrix}\beta_0\\ \beta_1\\ \vdots\\ \beta_{n-2}\\ \beta_{n-1}\end{bmatrix} \tag{7-75}$$

$$c=\left[0\ 0\cdots 0\ 1\right] 与 d=d' \tag{7-76}$$

【例 7-29】已知系统的传递函数为

$$\hat{G}(s)=\frac{3s^5+6s^4-10s^3+6s+2}{2s^5+4s^4-8s^3+2s+2}=1.5+\frac{s^3+1.5s-0.5}{s^5+2s^4-4s^3+s+1}$$

试求系统可控形及可观测形的动态方程实现。

【解】

由已知系统传递函数，根据式（7-73）～（7-76）可直接写出系统的可控标准形为：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\\ -1 & -1 & 0 & 4 & -2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 0\\ 0\\ 1\end{bmatrix}u$$

$$y=\begin{bmatrix}-0.5 & 1.5 & 0 & 1 & 0\end{bmatrix}x+1.5u$$

系统的可观测标准形为：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\\ \dot{x}_5\end{bmatrix}=\begin{bmatrix}0 & 0 & 0 & 0 & -1\\ 1 & 0 & 0 & 0 & -1\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 4\\ 0 & 0 & 0 & 1 & -2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\end{bmatrix}+\begin{bmatrix}-0.5\\ 1.5\\ 0\\ 1\\ 0\end{bmatrix}u$$

$$y=\begin{bmatrix}0 & 0 & 0 & 0 & 1\end{bmatrix}x+1.5u$$

7.5 系统状态反馈与状态观测器

负反馈闭环控制既是一种重要的控制策略，也是系统设计的主要方式。经典控制理论是用传递函数来描述系统的，常用输出量作反馈量，此即为输出反馈。现代控制理论的特点是用系统内部的状态变量来描述系统，所以在建立了状态变量的概念后，系统除了输出反馈之外，还有状态反馈这种新的控制形式，因此也要建立状态反馈的概念。

7.5.1 系统的输出反馈与状态反馈

先来介绍系统输出反馈后与状态反馈后的动态方程。

7.5.1.1 系统的输出反馈

设有如图 7-7 所示输出反馈系统，在原系统$\{A、B、C\}$中引入输出反馈后，系统的动态方程为：

$$\dot{x}=Ax+Bu-Hy \tag{7-77}$$

系统的输出方程为：

$$y=Cx \tag{7-78}$$

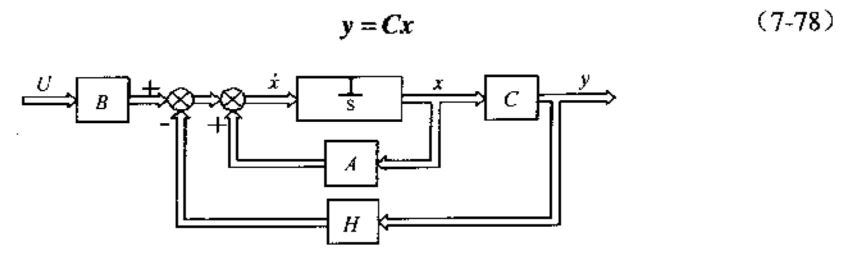


图 7-7 输出反馈系统

将式（7-77）代入式（7-78）即得：

$$\begin{cases}\dot{x}=(A-HC)x+Bu & (7\text{-}79)\\ y=Cx & (7\text{-}80)\end{cases}$$

这就是由输出反馈构成的新动态方程。

另一种输出反馈的形式如图 7-8 所示。图中$G(s)$为系统前向通道的传递矩阵，反馈通道的传递矩阵为$H(s)$。反馈向量$z(s)$与误差向量$e(s)$之间有：

$$z(s)=H(s)y(s)=H(s)G(s)e(s) \tag{7-81}$$

由此可求得$z(s)$与$e(s)$之间的传递矩阵并定义为：

$$\frac{z(s)}{e(s)}=H(s)G(s) \tag{7-82}$$

现在来求闭环系统的传递矩阵。由图 7-8 有

$$y(s)=G(s)[u(s)-z(s)]=G(s)[u(s)-H(s)y(s)]$$

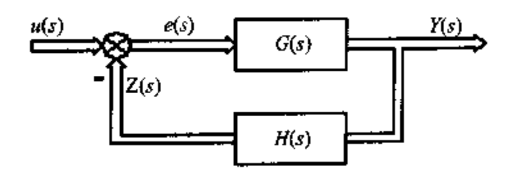


图 7-8 显示误差向量 $e(s)$的输出反馈系统

整理后可得

$$[I+G(s)H(s)]y(s)=G(s)u(s)$$

若 $\det[I+G(s)H(s)]\neq 0$， 用$[I+G(s)H(s)]^{-1}$左乘上式两边，有

$$y(s)=[I+G(s)H(s)]^{-1}G(s)u(s) \tag{7-83}$$

故得闭环系统的传递矩阵为：

$$\Phi(s)=\frac{y(s)}{u(s)}=[I+G(s)H(s)]^{-1}G(s) \tag{7-84}$$

7.5.1.2 系统的状态反馈

系统在采用现代控制理论后，常用状态反馈的方法来进行系统极点配置，使之具有特定的性能。

设系统的动态方程为：

$$\begin{cases} \dot{x}=Ax+Bu \\ y=Cx \end{cases}$$

x 为 n 维状态向量，u 为 p 维输入向量，y 为 q 维输出向量，A、B、C 分别为 $n\times n$、$n\times p$ 、$q\times n$ 矩阵。

图 7-9 所示为系统结构图，加入了状态反馈后的系统结构图如图 7-10 所示。

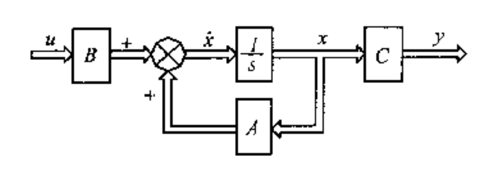


图 7-9　系统结构图

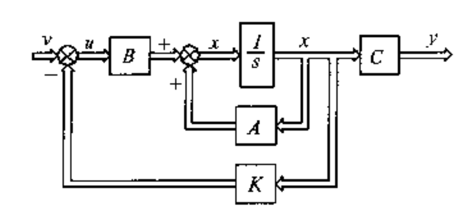


图 7-10　施加状态反馈后的系统结构图

图 7-10 中矩阵 K 叫做状态反馈矩阵，v 是闭环系统的输入，方框 B 的输入仍用 u 表示。因此引入状态反馈后，有：

$$u=v-Kx \tag{7-85}$$

此时闭环系统的状态方程可写作：

$$\dot{x}=Ax+Bu=Ax-BKx+Bv$$

或者

$$\dot{x}=(A-BK)x+Bv \tag{7-86}$$

式（7-86）中，$(A-BK)$为闭环系统的状态矩阵。于是，可以用系统$\{(A-BK),B,C\}$来表示引入状态反馈后的闭环系统，而输出方程没有变化，仍为：

$$y=Cx \tag{7-87}$$

7.5.2 系统的极点配置

不论是镇定系统还是跟随系统都必须符合规定性能指标的要求。控制系统的动态响应主要是山它的极点位置决定的。因此控制系统的设计就是应用状态反馈使闭环系统具有期望的极点配置，将原系统的开环极点变成期望的系统极点，从而改善系统性能。所

谓系统的极点配置问题，就是给定了闭环系统的极点位置后，如何通过某种方法来达到给定的极点配置。

对一个可控系统，在采用状态反馈后，可以实现闭环极点的任意配置。即是通过状态反馈的方法，使闭环系统的极点位于预先规定的位置上。由自动控制原理，实现闭环极点的任意配置的必要且充分条件是系统完全可控。

MATLAB 工具箱提供的函数 place（ ），它利用 Ackermann 公式计算反馈增益矩阵 K，使采用全反馈 $\boldsymbol{u}=-\boldsymbol{Kx}$ 的多输入系统具有指定的闭环极点 p。函数 place（ ） 的调用格式为：

$$\boldsymbol{K}=\text{place}\ (\boldsymbol{A},\boldsymbol{B},\boldsymbol{P})$$

其中，输入参量 $\boldsymbol{A}$ 为系统的状态矩阵，$\boldsymbol{B}$ 为系统的输入矩阵，$\boldsymbol{P}$ 为指定的闭环系统极点。返回参量 $\boldsymbol{K}$ 为反馈增益矩阵即状态反馈矩阵。

函数 place（ ）也可计算估计器增益矩阵 $\boldsymbol{K}$，使采用全反馈 $\boldsymbol{u}=-\boldsymbol{Kx}$ 的多输入系统具有指定的闭环极点 p。其调用格式为：

$$\boldsymbol{K}=\text{place}\ (\boldsymbol{A}',\boldsymbol{C}',\boldsymbol{P})$$

其中，输入参量 $\boldsymbol{A}$ 为系统的状态矩阵，$\boldsymbol{A}'$是 $\boldsymbol{A}$ 的转置矩阵；$\boldsymbol{C}$ 为系统的观测矩阵，$\boldsymbol{C}'$是 $\boldsymbol{C}$ 的转置矩阵；$\boldsymbol{P}$ 为指定的闭环系统极点。返回参量 $\boldsymbol{K}$ 为估计器增益矩阵。

【例 7-30】已知一系统的传递函数为

$$\frac{Y(s)}{U(s)}=\frac{10}{s(s+1)(s+2)}$$

试判别系统的可控性并设计反馈控制器，使得闭环系统的极点为-2 与 $(-1\pm i)$。

【解】

首先判别系统的可控性。给出以下程序 L7501a.m：

```
% MATLAB PROGRAM L7501a.m
%
n1=10;d1=conv(conv([1 0],[1 1]),[1 2]);
[a,b,c,d]=tf2ss(n1,d1);n=3;
CAM=ctrb(a,b);
if det(CAM)~=0
    rcam=rank(CAM)
  if rcam ==n
      disp('System is controlled')
  elseif rcam<n
     disp('System is no controlled')
  end
elseif det(CAM)==0
    disp('System is no controlled')
end
```

程序运行结论是：

```
System is controlled
```

由于系统是可控的，所以可任意配置系统的极点。为设计反馈控制器，给出以下调用函数 place （ ） 的程序 L7501b. m:

```
% MATLAB PROGRAM L7501b.m
%
n1=10;d1=conv(conv([1 0],[1 1]),[1 2]);
[a,b,c,d]=tf2ss(n1,d1);
p=[-2 -1+i -1-i];
k=place(a,b,p)
```

程序运行结果是：

```
place: ndigits= 15
k =
1.0000      4.0000      4.0000
```

所以状态反馈矩阵（反馈控制器）为：

$$\boldsymbol{k}=[1\ 4\ 4]$$

【例 7-31】已知一系统的状态方程

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}$$

方程中 A=[0 1 0 0;0 0 −1 0;0 0 0 1;0 0 11 0]; B=[0;1;0;−1]。试判别系统的可控性，并设计状态反馈控制器，使得闭环系统的极点为-1、-2、与$(-1\pm i)$。

【解】

根据题意，为判别系统的可控性并设计状态反馈控制器，给出以下调用函数place（ ）的程序 L7502.m:

```
% MATLAB PROGRAM L7502.m
%
A=[0 1 0 0;0 0 -1 0;0 0 0 1;0 0 11 0];
B=[0;1;0;-1];p=[-1 -2 -1+1i -1-1i];
CAM=ctrb(A,B);n=4;
if det(CAM)~=0
    rcam=rank(CAM)
  if rcam ==n
      disp('System is controlled')
  elseif rcam<n
      disp('System is no controlled')
  end
elseif det(CAM)==0
  disp('System is no controlled')
end
if rcam ==n
   k=place(A,B,p)
end
```

程序运行结果是系统完全可控：

```
System is controlled
place: ndigits= 15
k =
   -0.4000   -1.0000  -21.4000   -6.0000
```

所以反馈控制器为：

$$\boldsymbol{k}=[-0.4\ \ -1\ -21.4\ \ -6]$$

7.5.3 系统的状态观测器

为了实现状态反馈，需要系统全部的状态变量。但在实际系统中，大部分状态变量很难直接测量到。因此，为了实现状态反馈控制，需要通过一个模型，利用已知的信息（如输出量 y 与输入量 u）对系统状态变量进行估计。这样，可以构造一个与已知实际系统$\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}\}$具有同样动态方程的模拟系统，用模拟系统的状态向量$\hat{\boldsymbol{x}}(t)$作为实际系统状态向量$\boldsymbol{x}(t)$的估计值。状态观测器就是状态估计器。

一般情况下，系统的输出量$\boldsymbol{y}(t)$与控制输入量$\boldsymbol{u}(t)$均为已知，所以希望能从$\boldsymbol{y}(t)$与$\boldsymbol{u}(t)$对构造的系统模型来估计出状态变量。可以证明，当系统完全可观测时，一定存在状态观测器。状态观测器结构如图 7-11 所示。

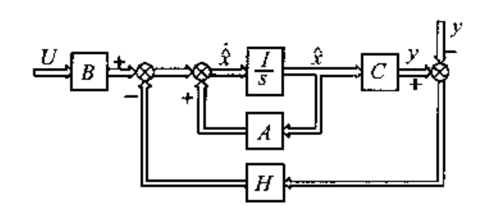


图 7-11　状态观测器结构

对图 7-11，有输出反馈后，观测器的状态方程为：

$$\dot{\hat{\boldsymbol{x}}}(t)=\boldsymbol{A}\hat{\boldsymbol{x}}(t)+\boldsymbol{B}\boldsymbol{u}-\boldsymbol{H}[\hat{\boldsymbol{y}}(t)-\boldsymbol{y}(t)] \quad (7\text{-}88)$$

对图 7-11 还有

$$\hat{\boldsymbol{y}}(t)=\boldsymbol{C}\,\hat{\boldsymbol{x}}(t) \quad (7\text{-}89)$$

故有

$$\dot{\hat{\boldsymbol{x}}}(t)=(\boldsymbol{A}-\boldsymbol{H}\boldsymbol{C})\hat{\boldsymbol{x}}(t)+\boldsymbol{B}\boldsymbol{u}(t)+\boldsymbol{H}\boldsymbol{y}(t) \quad (7\text{-}90)$$

将式（7-90）与原系统状态方程相减，可得状态向量误差方程如下：

$$[\dot{x}(t)-\dot{\hat{x}}(t)]=(\boldsymbol{A}-\boldsymbol{H}\boldsymbol{C})[\boldsymbol{x}(t)-\hat{\boldsymbol{x}}(t)] \quad (7\text{-}91)$$

在设计中，希望$\hat{\boldsymbol{x}}(t)$尽量快地趋近于$\boldsymbol{x}(t)$，则要合理地选择矩阵$(\boldsymbol{A}-\boldsymbol{H}\boldsymbol{C})$的特征值。带观测器的整个系统如图 7-12 所示。

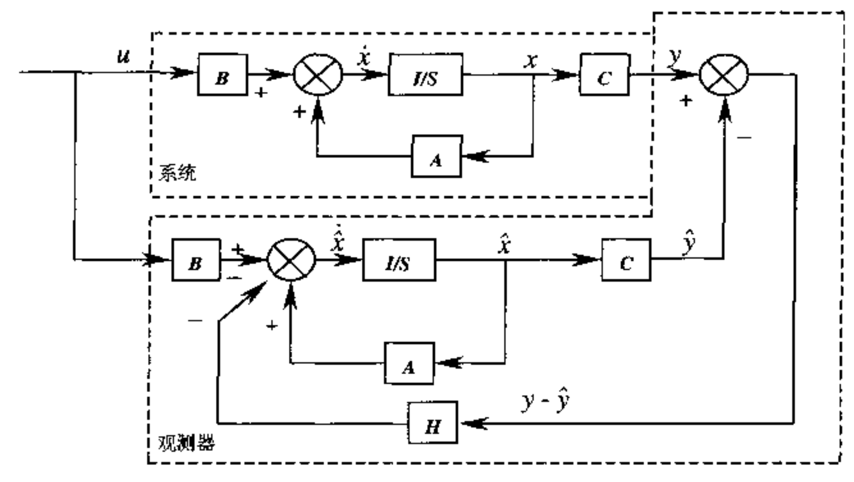


图 7-12　带观测器的系统

MATLAB 工具箱提供的 acker（ ）函数，它利用 Ackermann 公式计算估计器增益矩阵 $\boldsymbol{K}$，使采用全反馈 $\boldsymbol{u}=-\boldsymbol{Kx}$ 的单输入系统具有指定的闭环极点 p。

函数 acker（ ）的调用格式为：

K=acker（A',C',P）

其中输入参量 $\boldsymbol{A}$ 为系统的状态矩阵，$\boldsymbol{A}'$是 $\boldsymbol{A}$ 的转置矩阵；$\boldsymbol{C}$ 为系统的观测矩阵，$\boldsymbol{C}'$是 $\boldsymbol{C}$ 的转置矩阵；$\boldsymbol{P}$ 为指定的闭环系统极点。返回参量 $\boldsymbol{K}$ 为估计器增益矩阵，而 $\boldsymbol{K}'=\boldsymbol{H}$。

对于已知实际系统 $\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}\}$，通过函数 acker（ ）先求出 $\boldsymbol{K}$ 后，进而求得 $\boldsymbol{H}$，最后状态观测器式（7-90）即可解得。

【例 7-32】已知一系统的状态方程

$$\begin{cases} \dot{x}=ax+bu \\ y=cx \end{cases}$$

方程中 a=[0 1;-2 -3]；b=[0;1]；c=[2 0]。试判别系统的可观性并对系统设计状态观测器，使得闭环系统的极点为-10、-10。

【解】

（1）判别系统可观性

根据题意，给出以下程序 L7503a.m 判别系统的可观性:

```
% MATLAB PROGRAM L7503a.m
%
a=[0 1;-2 -3];b=[0;1];c=[2 0];
n=2;cb=obsv(a,c);
roam=rank(ob);
if roam ==n
    disp('System is observable')
elseif rcam~=n
disp('System is no observable')
end
```

程序运行后的结论是：

```
System is observable
```

（2）设计状态观测器

由于系统是可观的，所以存在状态观测器。根据式（7-90）给出调用函数 acker（）的程序 L7503b.m 用来设计状态观测器。

```
% MATLAB PROGRAM L7503b.m
%
a=[0 1;-2 -3];b=[0;1];c=[2 0];p1=[-10,-10];
a1=a';b1=c';c1=b';
k=acker(a1,b1,p1);h=(k)'
ahc=a-h*c
```

程序运行后的结果：

```
h =
    8.5000
   23.5000
ahc =
   -17       1
   -49      -3
```

即系统的状态观测器为：

$$\dot{\hat{x}}=\begin{bmatrix}-17 & 1\\ -49 & -3\end{bmatrix}\hat{x}+\begin{bmatrix}0\\ 1\end{bmatrix}u+\begin{bmatrix}8.5\\ 23.5\end{bmatrix}y$$

7.5.4 系统的状态降阶观测器

当观测器的阶数等于原系统状态向量的阶数时称观测器为全阶观测器。所谓状态降阶观测器就是观测器的阶数在低于原系统状态向量的阶数时，也能对原系统的状态做出最好的估计。

对于一个 n 维的可观测系统：

$$\dot{\bar{x}}=\bar{A}\bar{x}+\bar{B}u \tag{7-92}$$

$$\bar{y}=\bar{C}\bar{x} \tag{7-93}$$

式中，$\bar{x}$ 为 n 维向量，u 为 r 维向量，$\bar{y}$ 为 m 维向量，$\bar{A}$、$\bar{B}$、$\bar{C}$ 分别为 $n\times n$，$n\times r$，$m\times n$ 矩阵。当 $\bar{C}$ 满行秩（即 $\mathrm{rank}\,\bar{C}=m$）时，可以定义一种坐标变换，使系统 $\{\bar{A},\bar{B},\bar{C}\}$ 经变换后为系统 $\{A,B,C\}$，其动态方程将具有如下的典型分块矩阵形式：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}B_1\\ B_2\end{bmatrix}u \tag{7-94}$$

$$y=[0 \mid I]\begin{bmatrix}x_1\\ x_2\end{bmatrix}=x_2 \tag{7-95}$$

式中 x_1 为（n-m）维列向量；x_2 为 m 维列向量；

A_{11} 为（n-m）×（n-m）维矩阵；A_{12} 为（n-m）×m 维矩阵；

A_{21} 为 $m \times n$-m 维矩阵；A_{22} 为 $m \times m$ 维矩阵；

B_1 为（n-m）×m 维矩阵；B_2 为 $m \times r$ 维矩阵。

因为系统 $\{\overline{A},\overline{B},\overline{C}\}$ 可观测，经线性坐标变换后的系统 $\{A,B,C\}$ 仍为可观测系统。

由式（7-95）看出，通过直接测量输出 y，已可得到系统状态的一部分 x_2，现在只要设计出能估计状态 x_1 的（n-m）维观测器，它就是原系统 $\{A,B,C\}$ 的降阶观测器了。这个（n-m）维观测器通常叫做龙伯格观测器。

根据式（7-95），式（7-94）可以改写成如下形式：

$$\begin{cases}\dot{x}_1 = A_{11}x_1 + (A_{12}y + B_1u) \\ \dot{y} = A_{21}x_1 + A_{22}y + B_2u\end{cases} \tag{7-96}$$

或者

$$\begin{cases}\dot{x}_1 = A_{11}x_1 + (A_{12}y + B_1u) \\ y_1 = A_{21}x_1 = \dot{y} - A_{22}y - B_2u\end{cases} \tag{7-97}$$

式（7-97）表达的是以 x_1 为状态的子系统，其中的第一式是它的状态方程。其输入为 $(A_{12}y + B_1u)$，因为 y 可直接测量，故是已知的；第二式是输出方程，它表达了输出 y_1 与状态 x_1 的关系。

根据式（7-90），以 x_1 为状态的子系统即式（7-97）的状态观测器应为：

$$\dot{\hat{x}}_1 = (A_{11} - HA_{21})\hat{x}_1 + A_{12}\,y + B_1\,u + H\,y_1 \tag{7-98}$$

式中 H 为（n-m）×（m）维矩阵，观测器的输出就是 x_1。

考虑式（7-97），式（7-98）的降阶观测器可写成：

$$\dot{\hat{x}}_1 = (A_{11} - H\,A_{21})\hat{x}_1 + A_{12}y + B_1u + H\,\dot{y} - H(A_{22}y + B_2u) \tag{7-99}$$

式中 $\dot{y}$ 不是能直接量测得到的，为避免应用 $\dot{y}$，特作如下变量变换：

$$\begin{cases}z_1 = \hat{x}_1 - H\,y \\ \dot{z}_1 = \dot{\hat{x}}_1 - H\,\dot{y}\end{cases} \tag{7-100}$$

所以降阶观测器为：

$$\begin{cases}\dot{z}_1 = (A_{11} - HA_{21})z_1 + (B_1 - HB_2)u + [(A_{11} - HA_{21})H + A_{12} - HA_{22}]y \\ \hat{x}_1 = z_1 + H\,y\end{cases} \tag{7-101}$$

这样，系统 $\{A,B,C\}$ 的状态估计则按下式求得：

$$\hat{x} = \begin{bmatrix}\hat{x}_1 \\ x_2\end{bmatrix} = \begin{bmatrix}z_1 + Hy \\ y\end{bmatrix} = \begin{bmatrix}I \\ 0\end{bmatrix}z_1 + \begin{bmatrix}H \\ I\end{bmatrix}y \tag{7-102}$$

自动控制理论中已经说明，只要子系统（A_{11}，A_{21}）完全能观测，H 就一定存在。同样可以调用 acker（ ）函数计算估计器增益矩阵 K，由 $H=K'$，求出 H 后，状态观测器式（7-101）即可解得。

【例 7-33】已知一系统的状态方程为：

$$\begin{cases} \dot{x} = ax + bu \\ y = cx \end{cases}$$

方程中 ***a***=[0 1 0 0;0 0 -1 0;0 0 0 1;0 0 11 0]；***b***=[0;1;0;-1]；***c***=[1 0 0 0]；对系统：①试判别其可观性；②若系统可观，试设计全阶状态观测器，使得闭环系统的极点为-2、-3 与 $(-2 \pm i)$；③设计三阶状态观测器，使得闭环系统的极点为-3 与 $(-2 \pm i)$。

【解】

（1）判别系统可观性

根据题意，给出以下程序 L7504a.m 判别系统的可观性。

```
% MATLAB PROGRAM L7504a.m
%
a=[0 1 0 0;0 0 -1 0;0 0 0 1;0 0 11 0];b=[0;1;0;-1];c=[1 0 0 0];
n=4;ob=obsv(a,c);
roam=rank(ob);
if roam ==n
     disp('System is observable')
elseif rcam~=n
disp('System is no observable')
end
```

程序运行后的结论是：

```
System is observable
```

（2）设计全阶状态观测器

由于系统是可观的，所以存在状态观测器。给出调用函数 acker（）的程序 L7504b.m 用来设计全阶状态观测器。

```
% MATLAB PROGRAM L7504b.m
%
a=[0 1 0 0;0 0 -1 0;0 0 0 1;0 0 11 0];b=[0;1;0;-1];c=[1 0 0 0];
p1=[-2 -3 -2+i -2-i];a1=a';b1=c';c1=b';
k=acker(a1,b1,p1);
h=(k)'
ahc=a-h*c
```

程序运行后的结果是：

```
h =
     9
    42
  -148
  -492
ahc =
```

```
 -9     1     0     0
-42     0    -1     0
148     0     0     1
492     0    11     0
```

即全阶状态观测器为：

$$\dot{\hat{x}}=\begin{bmatrix}-9&1&0&0\\-42&0&-1&0\\148&0&0&1\\492&0&11&0\end{bmatrix}\hat{x}+\begin{bmatrix}0\\1\\0\\-1\end{bmatrix}u+\begin{bmatrix}9\\42\\-148\\-492\end{bmatrix}y$$

（3）设计三阶状态观测器

为便于设计三阶状态观测器，其状态变量表达式可改写如下：

$$\begin{bmatrix}\dot{x}_2\\\dot{x}_3\\\dot{x}_4\\\hline\dot{x}_1\end{bmatrix}=\left[\begin{array}{ccc|c}0&-1&0&0\\0&0&1&0\\0&11&0&0\\\hline1&0&0&0\end{array}\right]\begin{bmatrix}x_2\\x_3\\x_4\\\hline x_1\end{bmatrix}+\begin{bmatrix}1\\0\\-1\\\hline0\end{bmatrix}u$$

$$y=\left[\begin{array}{ccc|c}0&0&0&1\end{array}\right]\begin{bmatrix}x_2\\x_3\\x_4\\\hline x_1\end{bmatrix}=x_1$$

x_1 直接由 y 提供，可只设计三阶状态观测器。

根据改写的状态变量表达式与式（7-101），给出调用函数 acker（）的程序 L7504c.m 用来设计三阶状态观测器。

```
% MATLAB PROGRAM L7504c.m
%
a=[0 -1 0 0;0 0 1 0;0 11 0 0;1 0 0 0];b=[1;0;-1;0];c=[0 0 0 1];
a11=[a (1:3,1:3) ];a12=[a (1:3,4) ];a21=[a (4,1:3) ];a22=[a (4,4) ];
b1=b (1:3,1) ;b2=b (4,1) ;a1=a11;c1=a21;ax= (a1) ';bx= (c1) ';
p=[-3 -2+i -2-i];
k=acker (ax,bx,p) ;
h=k'
ahaz= (a11-h*a21)
bhbu=b1-h*b2
ahay= (a11-h*a21) *h+a12-h*a22
```

程序运行后的结果是：

```
h =
     7
   -28
   -92
```

```
ahaz =
    -7    -1     0
    28     0     1
    92    11     0
bhbu =
     1
     0
    -1
ahay =
   -21
   104
   336
```

即三阶状态观测器为：

$$\begin{cases} \dot{z}_1 = \begin{bmatrix} -7 & -1 & 0 \\ 28 & 0 & 1 \\ 92 & 11 & 0 \end{bmatrix} z_1 + \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} u + \begin{bmatrix} -21 \\ 104 \\ 336 \end{bmatrix} y \\ \hat{X}_2 = \begin{bmatrix} \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \end{bmatrix} = z_1 + \begin{bmatrix} 7 \\ -28 \\ -92 \end{bmatrix} y \end{cases}$$

7.6 系统稳定性分析的 MATLAB 计算及仿真

7.6.1 系统稳定性概述

在第四章 4.2 节控制系统稳定性分析的 MATLAB 实现中已经对经典控制理论的稳定性及其稳定判据作了介绍，并已指出，在状态空间分析里有用李雅普诺夫函数阐述稳定性的 Lyapunov 稳定概念。Lyapunov 稳定概念源于俄国数学家李亚普诺夫的稳定性理论，这个稳定性理论是分析系统稳定性方法的依据。

在经典控制理论分析中，稳定性的理论依据是控制系统特征方程式的根都分布在根（复）平面虚轴的左半部。这样由微分方程的时域解来判定系统稳定性的便是李亚普诺夫第一法。而现代控制理论中则用李亚普诺夫第二法（或叫直接法），直接法是指不用求解系统的运动方程直接可以判定系统稳定性的方法。因此对于不易求解的那些非线性系统与时变系统来说，李亚普诺夫直接法是特别有用的。

稳定性与可控性以及可观测性都是控制系统的重要属性。几乎所有可运行的系统必须设计成稳定的，不稳定的系统没有什么意义，因为它不能实际使用。系统稳定性是系统能够成立与运行的首要条件。

对于任何一个物理系统，都有运动稳定性的问题存在。所谓运动稳定性理论，就是研究某些确定性的或随机性的干扰作用对于运动状态的影响，从而建立判别运动状态是稳定的、渐近稳定的或是不稳定的准则。

为了说明李亚普诺夫的稳定性理论，先来介绍标量函数正定性、负定性与不定性等的概念。

1. 正定性

设有标量函数$\boldsymbol{V}(\boldsymbol{x})$，它对于 S 域中所有非零状态 $\boldsymbol{x}$（即$\boldsymbol{x}\neq\boldsymbol{0}$），总有$\boldsymbol{V}(\boldsymbol{x})>0$；且当$\boldsymbol{x}=0$（也在域 S 中）时，$\boldsymbol{V}(\boldsymbol{x})=0$，则称标量函数$\boldsymbol{V}(\boldsymbol{x})$在 S 域内是正定的。

2. 负定性

设有标量函数$\boldsymbol{V}(\boldsymbol{x})$，对于 S 域中所有非零状态$\boldsymbol{x}$，总有$\boldsymbol{V}(\boldsymbol{x})<0$；且$V(0)=0$，则称标量函数$\boldsymbol{V}(\boldsymbol{x})$在域 S 内是负定的。$-\boldsymbol{V}(\boldsymbol{x})$则是正定的。

3. 半正定性与半负定性

在域 S 中，对$(\boldsymbol{A}-\boldsymbol{BK})$及某些状态$\boldsymbol{x}$，$\boldsymbol{V}(\boldsymbol{x})=0$；对 S 中所有其他状态，都有$\boldsymbol{V}(\boldsymbol{x})>0$，则称$\boldsymbol{V}(\boldsymbol{x})$是半正定的。如果$-\boldsymbol{V}(\boldsymbol{x})$是半正定的，则$\boldsymbol{V}(\boldsymbol{x})$是半负定的。

4. 不定性

不管域 S 多么小，在域 S 内，$\boldsymbol{V}(\boldsymbol{x})$ 能正能负，则称$\boldsymbol{V}(\boldsymbol{x})$是不定的。

5. 非负定性

正定性与半正定性就是非负定性。

7.6.2 李亚普诺夫意义下的稳定性理论

1. 设系统的状态方程为：

$$\dot{\boldsymbol{x}}=f(\boldsymbol{x},t) \tag{7-103}$$

其平衡状态为：

$$f(0,t)=0 \tag{7-104}$$

如果存在一个具有连续一阶偏导数的标量函数$V(\boldsymbol{x},t)$，并且满足以下条件：

（1）$\boldsymbol{V}(\boldsymbol{x},\boldsymbol{t})$ 是正定的；

（2）$\dot{\boldsymbol{V}}(\boldsymbol{x},\ \boldsymbol{t})$是负定的，

则系统在状态空间原点处的平衡状态是一致渐近稳定的。

如果当$\|\boldsymbol{x}\|\to\infty$时，有$V(\boldsymbol{x},t)\to\infty$，则系统在状态空间原点处的平衡状态是在大范围内一致渐近稳定的。

满足以上两个条件的标量函数叫做李亚普诺夫函数。李亚普诺夫函数的几何意义是：$V(\boldsymbol{x})$表示状态空间的原点到$\boldsymbol{x}$之间的距离，而$\dot{V}(\boldsymbol{x})$则代表在$\boldsymbol{x}$点处趋向状态空间原点的速度。

2. 设系统的状态方程为：

$$\dot{\boldsymbol{x}}=f(\boldsymbol{x},\ t)$$

其平衡状态为：

$$f(0,\ t)=0\quad(t\geqslant t_0) \tag{7-105}$$

如果存在一个具有连续一阶偏导数的标量函数$\boldsymbol{V}(\boldsymbol{x},\boldsymbol{t})$，并且满足以下条件：

（1）$V(\boldsymbol{x},\ t)$ 是正定的；

（2）$\dot{V}(\boldsymbol{x},\ t)$ 是半负定的；

则系统在状态空间原点的平衡状态是稳定的。

3. 设系统的状态方程为：

$$\dot{x} = f(x,\ t)$$

其平衡状态为：

$$f(0,\ t) = 0 \quad (t \geqslant t_0)$$

如果存在一个具有连续一阶偏导数的标量函数 $W(x,\ t)$，并且满足以下条件：

（1）$W(x,\ t)$ 在原点的某一邻域内是正定的；

（2）$\dot{W}(x,\ t)$ 在同样的邻域内也是正定的，

则系统在状态空间原点的平衡状态是不稳定的。

7.6.3 线性定常系统的李亚普诺夫稳定性分析

设线性定常系统为：

$$\dot{x} = Ax \tag{7-106}$$

式中，x 为 n 维状态向量，A 为 $n \times n$ 常系数矩阵。假定 A 为非奇异矩阵，则系统的唯一平衡状态在状态空间原点 $x = 0$ 处。

对于方程（7-106）所确定的系统，取一个可能的标量函数：

$$V(x) = x^T P x \tag{7-107}$$

式中，P 是一个正定的实对称矩阵。

将 $V(x)$ 对于时间求导，可得到：

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T P x + x^T P \dot{x} = (Ax)^T P x + x^T P A x = \\ & x^T A^T P x + x^T P A x = x^T (A^T P + PA) x \end{aligned} \tag{7-108}$$

由于 $V(x)$ 取为正定的，故若 $V(x)$ 为李亚普诺夫函数，则 $\dot{V}(x)$ 必为负定的，即有：

$$\dot{V}(x) = \ \ x^T Q x \tag{7-109}$$

上式中要求 Q 为正定阵，即

$$Q = \ \ (A^T P + PA) \tag{7-110}$$

由上可见：系统 $\dot{x} = Ax$ 在平衡状态 $x = 0$ 是大范围渐近稳定的充要条件为：给定一个正定的实对称矩阵 Q（通常选 Q 为单位矩阵），存在一个正定的实对称矩阵 P，使得

$$A^T P + PA = -Q \tag{7-111}$$

成立。这个结论是一个非常重要的定理，利用李亚普诺夫函数确定系统的稳定性时大多要用到这个定理。

标量函数 $V(x) = x^T P x$ 就是一个李亚普诺夫函数。有些文献把式（7-111）叫做李亚普诺夫方程。在 MATLAB 中，提供了对李亚普诺夫方程求解的函数。

7.6.4 李亚普诺夫方程的求解

在 MATLAB 工具箱中，提供了求解李亚普诺夫方程的函数 lyap（）。函数 lyap（）的调用格式为：

x=lyap（*A*,*C*）

其中，输入参量 ***A*** 是已知系统的状态矩阵，***C*** 就是给定的正定实对称矩阵 ***Q***。返回的输出变量 ***x*** 就是李亚普诺夫方程的解即正定实对称矩阵 ***P***。

求解李亚普诺夫方程的函数还有 dlyap（）与 lyap2（）。函数 dlyap（）用于求解离散系统的李亚普诺夫方程。函数 lyap2（）是采用特征值分解技术来求解李亚普诺夫方程，其运算速度比函数 lyap（）快很多。它们的调用格式为：

x=dlyap（*a*,*c*）

x=lyap2（*a*,*c*）

【例 7-34】已知线性定常系统如图 7-13 所示。

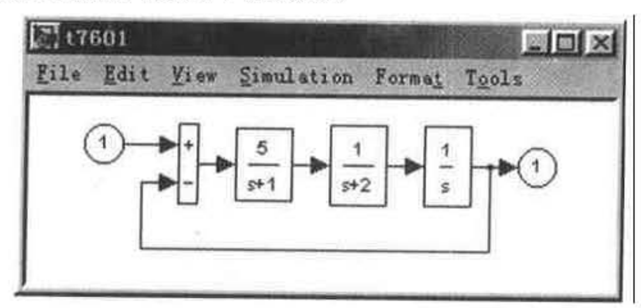


图 7-13 已知的线性定常系统

试求系统的状态方程；选择正定的实对称矩阵 *Q* 后计算李亚普诺夫方程的解并利用李亚普诺夫函数确定系统的稳定性。

【解】

讨论系统稳定性时可令给定输入 ***u***=0。根据题目要求，因为需调用函数 lyap（），故首先要将系统转换成状态空间模型。选择半正定实对称矩阵 ***Q*** 为：

$$Q=\begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}$$

为了确定系统的稳定性，需验证 ***P*** 阵的正定性，这可以对各主子行列式进行校验。综合以上考虑，给出调用函数 lyap（）的程序 L7601.m。

```
% MATLAB PROGRAM L7601.m
%
n1=5;d1=[1 1];s1=tf (n1,d1) ;
n2=1;d2=[1 2];s2=tf (n2,d2) ;
n3=1;d3=[1 0];s3=tf (n3,d3) ;
s123=s1*s2*s3;sb=feedback (s123,1) ;
[a]=tf2ss (sb.num{1},sb.den{1})
```

```
q=[0 0 0;0 0 0;0 0 1];
if det (a) ~=0
    p=lyap (a,q)
    det1=det (p (1,1))
    det2=det (p (2,2))
    detp=det (p)
end
```

运行程序后可得：

```
a =
    -3       -2       -5
     1        0        0
     0        1        0
p =
    12.5000      -0.0000      -7.5000
    -0.0000      7.5000       -0.5000
    -7.5000      -0.5000      4.7000
det1 =
    12.5000
det2 =
    7.5000
detp =
    15.6250
```

即系统的状态方程为：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -3 & -2 & -5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

李亚普诺夫方程的解为：

$$\boldsymbol{P} = \begin{bmatrix} 12.5 & 0 & -7.5 \\ 0 & 7.5 & -0.5 \\ -7.5 & -0.5 & 4.7 \end{bmatrix}$$

因为

$$\boldsymbol{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

是半正定阵，由式（7-109）有：

$$\dot{V}(\boldsymbol{x}) = -\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} = -x_3^2$$

是半负定的。最后，对各主子行列式（det1、det2、detp）进行校验说明 $\boldsymbol{P}$ 阵确是正定阵，因此本系统在坐标原点的平衡状态是稳定的，而且是大范围渐近稳定的。

【例 7-35】已知线性系统动态方程为：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ -1 & -1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

试计算李亚普诺夫方程的解；利用李亚普诺夫函数确定系统的稳定性并求李亚普诺夫函数。

【解】

首先选择正定实对称矩阵 $\boldsymbol{Q}$ 为单位矩阵，即

$$\boldsymbol{Q}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

根据题意，给出调用函数 lyap（）的程序 L7602.m。

```
% MATLAB PROGRAM L7602.ms
%
a=[0 1;-1 -1];q=[1 0;0 1];
if det (a) ~=0
    p=lyap (a,q)
    det1=det (p (1,1))
    detp=det (p)
end
```

运行程序后可得：

```
p =
   1.5000    -0.5000
  -0.5000     1.0000
det1 =
   1.5000
detp =
   1.2500
```

即李亚普诺夫方程的解为：

$$\boldsymbol{P}=\begin{bmatrix}1.5 & -0.5\\ -0.5 & 1\end{bmatrix}$$

程序已对各主子行列式（det1、detp）进行计算，计算结果说明 P 阵确是正定阵。

根据式（7-107），系统的李亚普诺夫函数为：

$$\boldsymbol{V}(x)=x^T Px=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}1.5 & -0.5\\ -0.5 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\frac{1}{2}\left(3x_1^2+2x_1x_2+2x_2^2\right)$$

在状态空间内，$\boldsymbol{V}(\boldsymbol{x})$是正定的，而

$$\dot{V}(x)=x^T\left(\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}\right)x=x^T(-I)x=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=-\left(x_1^2+x_2^2\right)$$

在状态空间内，$\dot{V}(\boldsymbol{x})$是负定的。另有，当$\|\boldsymbol{x}\|\to\infty$时，有$V(\boldsymbol{x},t)\to\infty$，因此系统原点处的平衡状态是大范围内渐近稳定的。

第 8 章　线性二次型最优控制的 MATLAB 实现

线性二次型最优控制是一种普遍采用的最优控制系统设计方法，本章共有 5 节，主要介绍线性二次型最优控制的 MATLAB 实现。8.1 节复习有关最优控制的基本概念；8.2 节介绍连续系统线性二次型最优控制的 MATLAB 实现；8.3 节介绍离散系统线性二次型最优控制的 MATLAB 实现；8.4 介绍最优观测器的 MATLAB 实现；8.5 节介绍线性二次型 Guass 最优控制的 MATLAB 实现。

8.1　最优控制的基本概念

正如第六章所介绍的采用经典控制理论设计控制系统，是按照给定的传统时域性能指标与频域性能指标要求来设计系统与进行校正。以最优控制的观点，首先，这样的方法是不严格的，多数是建立在凭经验的试凑基础上。而且往往在设计开始时就确定了系统的结构形式，而这些结构形式通常不是最优的。其次，在经典反馈控制中，只用系统的输入与输出信号差作为激励信号，对于比较复杂的控制系统，这种形式提供的信息不足以给出最优控制。因此，应用经典理论设计多输入-多输出系统与阶次较高的系统，往往得不到满意的结果。虽然用经典理论设计控制系统是基础，能够解决许多简单、确定系统的实际设计问题，但是对于诸多新型而复杂的控制系统设计还需要有在状态空间模型下建立的的最优控制策略。

所谓最优控制系统，就是在一定的具体条件下，在完成所要求的控制任务时，系统的某种性能指标具有最优值。根据系统不同的用途，可提出各种不同的性能指标。最优控制系统的设计，就是选择最优控制，以使某一种性能指标为最小。最优控制是现代控制理论的核心。

为了介绍线性二次型最优控制及其 MATLAB 的实现，有必要简要复习有关系统最优控制的基本问题与相关概念。

8.1.1　最优控制问题

设系统状态方程为：

$$\dot{x}(t)=f[x(t),u(t),\ t],x(t_0)=x_0 \tag{8-1}$$

式中，$x(t)$是 n 维状态向量；$u(t)$是 r 维控制向量；n 维向量函数 $f[x(t),u(t),t]$ 是 $x(t)$、$u(t)$与 t 的连续函数，且对 $x(t)$与 t 连续可微；$u(t)$在$[t_0,\ t_f]$上分段连续。所谓最优控

制问题，就是要求寻找最优控制函数，使得系统状态$\boldsymbol{x}(t)$从已知初态$\boldsymbol{x}_0$转移到要求的终端状态$\boldsymbol{x}(t_f)$，在满足如下约束条件下：

（1）控制与状态的不等式约束

$$g[\boldsymbol{x}(t),\boldsymbol{u}(t),t]\geqslant 0 \tag{8-2}$$

（2）终端状态的等式约束

$$M[\boldsymbol{x}(t_f),t_f]=0 \tag{8-3}$$

使性能指标

$$J=\Theta[\boldsymbol{x}(t_f),\ t_f]+\int_{t_0}^{t_f}F[\boldsymbol{x}(t),\boldsymbol{u}(t),t]\mathrm{d}t \tag{8-4}$$

达到极值。式中，$g[\boldsymbol{x}(t),\boldsymbol{u}(t),\ t]$是$m$维连续可微的向量函数，$m\leqslant r$；$M[\boldsymbol{x}(t_f),t_f]$是$s$维连续可维的向量函数，$s\leqslant n$；$\Theta[\boldsymbol{x}(t_f),t_f]$与$F[\boldsymbol{x}(t),\boldsymbol{u}(t),t]$都是$\boldsymbol{x}(t)$与$t$的连续可微纯量函数。

8.1.2 最优控制的性能指标

自动控制的性能指标是衡量系统性能好坏的尺度，其内容与形式取决于最优控制问题所要完成的任务，不同的控制问题应取不同的性能指标，其基本类型如下。

（1）积分型性能指标

$$J=\int_{t_0}^{t_f}F[\boldsymbol{x}(t),\boldsymbol{u}(t),t]\mathrm{d}\,t \tag{8-5}$$

表示在整个控制过程中，状态$\boldsymbol{x}(t)$与控制$\boldsymbol{u}(t)$应达到某些要求。例如：

①最小时间控制

取 $$F[\boldsymbol{x}(t),\boldsymbol{u}(t),t]=1$$

则

$$J=\int_{t_0}^{t_f}\mathrm{d}t=t_f-t_0 \tag{8-6}$$

②最小燃料消耗控制

取 $$F[\boldsymbol{x}(t),\boldsymbol{u}(t),t]=|\boldsymbol{u}(t)|$$

则

$$J=\int_{t_0}^{t_f}|\boldsymbol{u}(t)|\mathrm{d}\,t \tag{8-7}$$

③最小能量控制

取 $$F[\boldsymbol{x}(t),\ \boldsymbol{u}(t),\ t]=\boldsymbol{u}^2(t)$$

则

$$J=\int_{t_0}^{t_f}\boldsymbol{u}^2(t)\mathrm{d}\,t \tag{8-8}$$

④无限时间线性调节器

取$t_f \to \infty$，且

$$F(x,u,t)=\frac{1}{2}\left[x^T(t)Qx(t)+u^T(t)Ru(t)\right]$$

其中，$Q \geqslant 0$，$R > 0$，均为加权矩阵，则

$$J=\frac{1}{2}\int_{t_0}^{\infty}\left[x^T(t)Qx(t)+u^T(t)Ru(t)\right]dt \tag{8-9}$$

⑤无限时间线性跟踪器

$$J=\frac{1}{2}\int_{t_0}^{\infty}\left\{[y(t)-z(t)]^T Q[y(t)-z(t)]+u^T(t)Ru(t)\right\}dt \tag{8-10}$$

其中，$y(t)$为系统输出向量，$z(t)$为系统希望输出向量。

在性能指标式（8-8）、（8-9）、（8-10）中，被积函数都是$x(t)$、$y(t)-z(t)$或$u(t)$的平方项所组成，这种形式的性能指标叫做二次型性能指标。

（2）末值型性能指标

$$J=\Theta\left[x(t_f),t_f\right] \tag{8-11}$$

表示系统在控制过程结束后，要求系统的终端状态$x(t_f)$应达到某些要求，在实际工程中，例如要求导弹的脱靶量最小、机床工作台移动准确停止等。终端时刻t_f可以固定，也可以自由，视最优控制问题的性质而定。

（3）复合型性能指标

$$J=\Theta\left[x(t_f),t_f\right]+\int_{t_0}^{t_f}F[x(t),u(t),t]dt \tag{8-12}$$

表示对控制过程及控制过程结束后的终端状态均有要求，是最一般的性能指标形式。

8.1.3 最优控制问题的数学分类法

最优控制过程从数学意义上讲，还有以下的分类讲究。

（1）拉格朗日问题

如果已知系统的状态方程

$$\dot{x}(t)=f[x(t),u(t),t]$$

与初始状态$x(t_0)=x_0$，要求最优控制$u(t)$，使得以下性能指标

$$J=\int_{t_0}^{t_f}F[x(t),u(t),t]dt$$

为极小。这样的最优控制问题叫做拉格朗日问题。以上介绍的积分型性能指标所涉及的五种最优控制问题都属于拉格朗日问题。

（2）迈耶尔问题

如果已知系统的状态方程

$$\dot{x}(t)=f[x(t),u(t),t]$$

与初始状态$x(t_0)=x_0$，但是终端状态的某些分量$x(t_f)$与终端时刻t_f没有给定，要求最优控制$u(t)$，使得以下性能指标

$$J=\Theta[x(t),u(t),t]\Big|_{t_0}^{t_f} \tag{8-13}$$

为极小。这样的最优控制问题叫做迈耶尔问题。迈耶尔问题是一个终端量最小问题。以上介绍的末值型性能指标所涉及的最优控制问题都属于迈耶尔问题。

（3）波尔扎问题

已知系统的状态方程

$$\dot{x}(t)=f[x(t),u(t),t]$$

要求最优控制$u(t)$，使得以下性能指标

$$J=\Theta[x(t),u(t),t]_{t_0}^{t_f}+\int_{t_0}^{t_f}F[x(t),u(t),t]\mathrm{d}t \tag{8-14}$$

为极小。波尔扎问题是前述两个问题的复合。

这三类问题可以通过适当的变换相互转换。求解最优控制问题的基本数学原理有变分法、极小值原理与动态规划法。

8.1.4 最优控制问题的求解方法

（1）解析法

当性能指标与约束条件为显式解析表达式时，适合用解析法。通常是用求导方法或变分方法求出最优控制的必要条件，从而得到一组方程式或不等式，然后求解这组方程或不等式，最后得到最优控制的解析解。

（2）数值计算法

当性能指标比较复杂或不能用变量的显函数表示时，可以采用试探法，即直接搜索逐步逼近，经过若干次迭代，逐步逼近到最优点。

（3）梯度型法

这是一种解析与数值计算相结合的方法。

8.1.5 线性二次型最优控制

在介绍线性二次型最优控制之前，对最优控制的有关基本概念做以上的简要回顾是很有必要的。

对于线性系统，若性能指标为二次型函数，这样实现的控制叫做线性二次型最优控制。线性二次型最优控制方法是 20 世纪 60 年代发展起来的一种普遍采用的最优控制系统设计方法。这种方法的对象是以状态空间表达式给出的线性系统，而性能指标（或叫目标函数）为对象状态与控制输入的二次型函数。二次型问题就是在线性系统的约束条件下，选择控制输入使得二次型目标函数达到最小。

到目前为止，这种二次型最优控制在理论上比较成熟，为解决这类控制问题而开发的 MATLAB 函数也比较多，而且这种控制应用非常广泛。本章主要介绍线性二次型最优控制的 MATLAB 实现。

8.2 连续系统线性二次型最优控制的 MATLAB 实现

8.2.1 连续系统线性二次型最优控制

设线性连续定常系统的状态方程为：

$$\dot{x}(t)=Ax(t)+Bu(t),\quad x(0)=x_0 \tag{8-15}$$

式中，$x(t)$为 n 维状态向量：$u(t)$为 r 维控制向量，且不受约束；A 为 $n\times n$ 维常数矩阵，B 为 $n\times r$ 维常数矩阵。

系统的性能指标为：

$$J=\frac{1}{2}\int_0^\infty\left(x^TQx+u^TRu\right)dt \tag{8-16}$$

式中，终端时间 t_f 无限；Q 为 $n\times n$ 维常数矩阵；R 为 $r\times r$ 常数矩阵；$R>0$，$R=R^T$。若下列条件之一满足：

（1）$Q>0$，$Q=Q^T$，阵对 $[A,B]$ 完全可控；

（2）$Q\geq 0$，$Q=Q^T$，阵对 $[A,B]$ 完全可控，阵对 $[A,D]$ 完全可观，$DD^T=Q$，D 为任意矩阵，则有最优反馈矩阵：

$$K=R^{-1}B^TP \tag{8-17}$$

与唯一的最优控制：

$$u^*(t)=-Kx(t)=-R^{-1}B^TPx(t) \tag{8-18}$$

以及最优性能指标：

$$J^*=\frac{1}{2}x^T(0)Px(0) \tag{8-19}$$

式中，P 为常值正定矩阵，它是以下黎卡提代数方程的唯一解：

$$PA+A^TP-PBR^{-1}B^TP+Q=0 \tag{8-20}$$

闭环系统：

$$\dot{x}(t)=\left(A-BR^{-1}B^TP\right)x(t),\quad x(0)=x_0 \tag{8-21}$$

是渐近稳定的，其解为最优轨线 $x^*(t)$。

有些文献，把以上的线性二次型(Linear Quadratic)最优控制叫做 LQ 最优控制或者称为无限长时间定常系统的状态调节器控制。

8.2.2 连续系统线性二次型最优控制的 MATLAB 实现

在 MATLAB 的工具箱里，有特别提供的函数来求解连续系统线性二次型状态调节器问题。其函数有 lqr ( )、lqr2 ( ) 与 lqry ( )。函数的调用格式为：

[*K*,*S*,*E*]=lqr(*A*,*B*,*Q*,*R*,*N*)

[*K*,*S*]=lqr2(*A*,*B*,*Q*,*R*,*N*)

[K,S,E]=lqry(*sys*,**Q**,**R**,**N**)

其中，输入参量 *sys* 为系统的模型；**A** 为系统的状态矩阵；**B** 为系统的输入矩阵；**Q** 为给定的半正定实对称矩阵；**R** 为给定的正定实对称矩阵；**N** 代表更一般化性能指标中交叉乘积项的加权矩阵。输出参量 **K** 为最优反馈增益矩阵；**S** 为对应 Riccati 方程的唯一正定解 **P**（若矩阵 **A-BK** 是稳定矩阵，则总有 **P** 的正定解存在）；**E** 为 **A-BK** 的特征值。

函数 lqr2 ( ) 与 lqr ( ) 函数类似，只是其计算采用了 Schar 方法，因此具有更强的鲁棒性。

函数 lqry ( ) 用来求解二次型状态调节器的特例，是用输出反馈替代状态反馈，即有：

$$\boldsymbol{u}(t) = -\boldsymbol{K}\,\boldsymbol{y}(t) \tag{8-22}$$

其性能指标则为：

$$J = \frac{1}{2}\int_0^\infty \left(\boldsymbol{y}^T \boldsymbol{Q}\,\boldsymbol{y} + \boldsymbol{u}^T \boldsymbol{R}\,\boldsymbol{u}\right)\mathrm{d}\,t \tag{8-23}$$

这种二次型输出反馈控制叫做次优（或准最优）控制。

【例 8-1】已知系统动态方程为：

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\boldsymbol{u}$$

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\boldsymbol{x}$$

系统结构如图 8-1 所示。

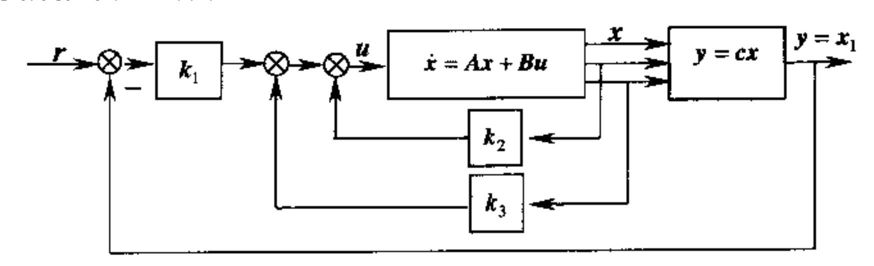


图 8-1　系统结构图

由结构图有系统的控制信号：

$$\boldsymbol{u} = k_1(\boldsymbol{r} - \boldsymbol{x}_1) - (k_2\boldsymbol{x}_2 + k_3\boldsymbol{x}_3) = k_1\boldsymbol{r} - (k_1\boldsymbol{x}_1 + k_2\boldsymbol{x}_2 + k_3\boldsymbol{x}_3) = k_1\boldsymbol{r} - \boldsymbol{K}\,x$$

式中反馈增益矩阵 **K**：

$$\boldsymbol{K} = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}$$

系统性能指标为：

$$J = \frac{1}{2}\int_0^\infty \left(\boldsymbol{x}^T \boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R}\boldsymbol{u}\right)\mathrm{d}\,t$$

其中

$$\boldsymbol{Q} = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\quad \boldsymbol{R} = \begin{bmatrix} 1 \end{bmatrix}$$

试计算最优状态反馈矩阵 $K=[k_1 \quad k_2 \quad k_3]$ 使 J 最小并对其闭环系统进行单位阶跃给定响应的仿真。

【解】

根据题意，须首先求解出最优反馈矩阵 K，这可调用 lqr( )函数来解决。由系统结构图 8-1，系统的状态方程为：

$$\dot{x}=Ax+Bu=Ax+B(\mathrm{k}_1 r-Kx)=(A-BK)x+B\,\mathrm{k}_1 r$$

系统的输出方程不变。根据这些考虑，给出以下程序 L8201.m。

```
% MATLAB PROGRAM L8201.m
%
a=[0 1 0;0 0 1;-6 -11 -6]; b=[0;0;1];
c=[1 0 0];d=[0];Q=[1000 0 0;0 1 0;0 0 1];  R=[1];
K=lqr(a,b,Q,R)
k1=K(1);ac=a-b*K;bc=b*k1;
cc=c;dc=d;
step(ac,bc,cc,dc)
```

运行程序后求出最优状态反馈矩阵 K 及其如图 8-2 所示的闭环系统单位阶跃给定响应的仿真曲线。

```
K = 26.1870   12.6189   1.8891
```

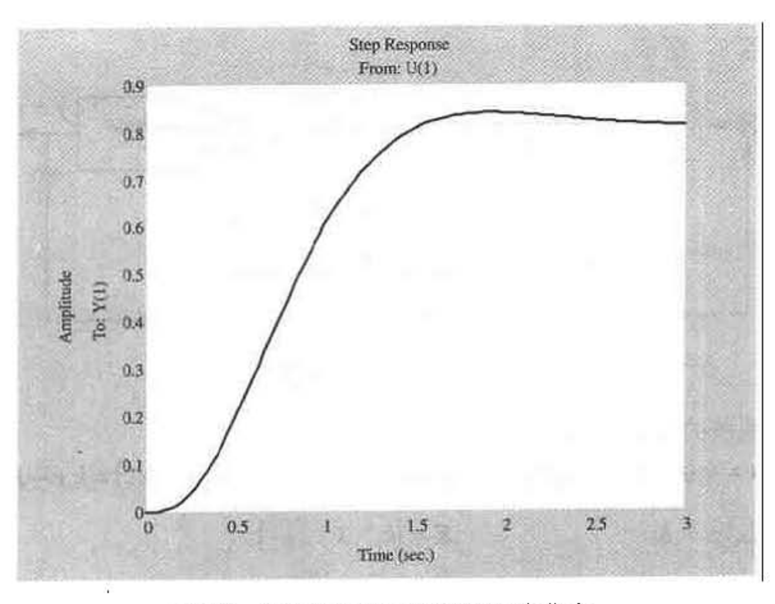


图 8-2 闭环系统单位阶跃给定响应曲线

经状态最优反馈后，闭环系统单位阶跃给定响应略微超调后立即单调衰减，这样的仿真曲线是很理想的，确实反映了最优控制的效果。

【例 8-2】续上题。要求采用输出反馈，即

$$u(t)=-K\,y(t)$$

使性能指标为：

$$J=\frac{1}{2}\int_0^\infty\left(y^TQ\,y+u^TRu\right)\mathrm{d}\,t$$

其中
$$Q=\begin{bmatrix}1000 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix},\quad R=[1]$$

试计算最优输出反馈矩阵 $K=\begin{bmatrix}k_1 & k_2 & k_3\end{bmatrix}$ 使 J 最小并对其闭环系统进行单位阶跃给定响应的仿真。

【解】

根据题意，求解最优输出反馈矩阵 K 须调用函数 lqry()。依照上题，特给出以下程序 L8202.m。

```
% MATLAB PROGRAM L8202.m
%
a=[0 1 0;0 0 1;-6 -11 -6]; b=[0;0;1];
c=[1 0 0];d=[0];Q=diag([1000]);R=[1];
K=lqry(a,b,c,d,Q,R)
k1=K(1);ac=a-b*K;bc=b*k1;
cc=c;dc=d;
step(ac,bc,cc,dc)
```

运行程序后求出最优输出反馈矩阵 K 及其如图 8-3 所示的闭环系统单位阶跃给定响应的仿真曲线。

```
K =
   26.1870   12.4878   1.8087
```

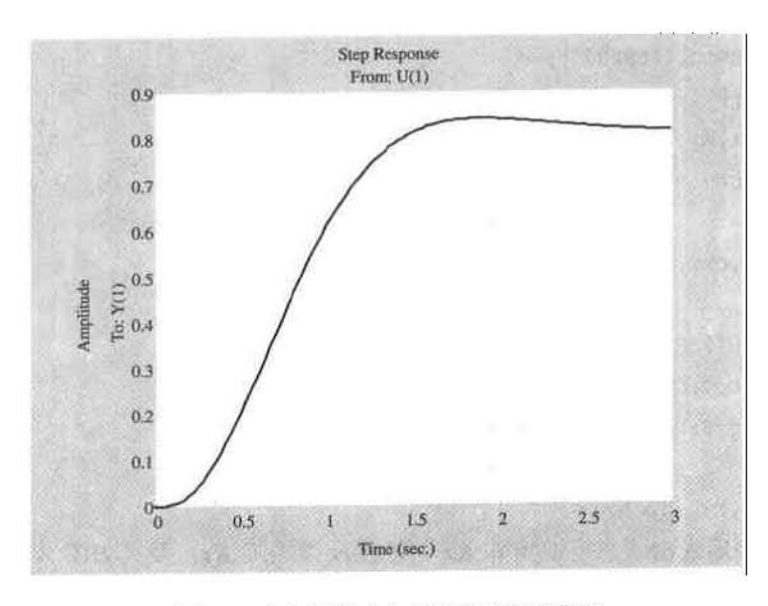


图 8-3　闭环系统单位阶跃给定响应曲线

对比图 8-2 与图 8-3 可见，经最优输出反馈后，闭环系统阶跃给定响应曲线与经状态反馈后的阶跃响应曲线相差无几。

【例 8-3】已知可控直流电源供电给直流电机的系统结构图如图 8-4 所示。欲对系统进行最优状态反馈与输出反馈控制，试分别计算状态反馈增益矩阵与输出反馈增益矩阵，并对其闭环系统进行阶跃响应仿真。

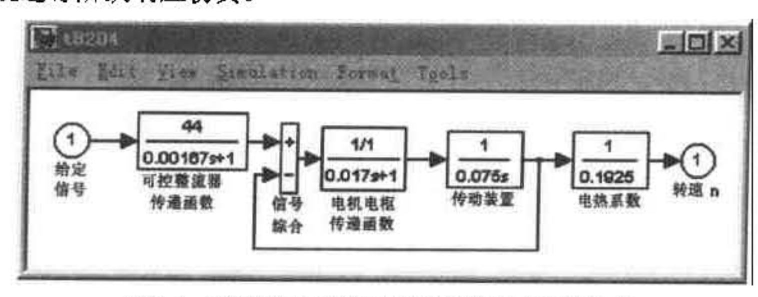


图 8-4　可控直流电源供电给直流电机系统结构图

给定状态反馈时

$$Qx=\begin{bmatrix}1000 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix},\quad R=[1]$$

给定状态反馈时

$$Qy=\mathrm{diag}([1000]);\ R=[1]$$

【解】

根据题意，要将已知系统结构图模型转换成状态空间模型，需要调用函数 linmod()。仿照以上两例，给出程序 L8203.m。

```
% MATLAB PROGRAM L8203.m
%
[a,b,c,d]=linmod2('t8204');
Qx=[1000 0 0;0 1 0;0 0 1];R=[1];
Kx=lqr(a,b,Qx,R)
kx1=Kx(1);axc=a-b*Kx;bxc=b*k1;cxc=c;dxc=d;
figure(1);
step(axc,bxc,cxc,dxc);hold on
figure(2);
Qy=diag([1000]);R=[1];
Ky=lqry(a,b,c,d,Qy,R)
ky1=Ky(1);ayc=a-b*Ky;byc=b*ky1;
cyc=c;dyc=d;
step(ayc,byc,cyc,dyc)
```

运行程序后求出状态反馈矩阵 Kx 与输出反馈矩阵 Ky，及其如图 8-5 所示采用状态反馈的闭环系统单位阶跃给定响应的仿真曲线与如图 8-6 所示采用输出反馈的闭环系统单位阶跃给定响应的仿真曲线。

```
Kx = 31.1506   337.0313   9.8154
Ky = 1.0e+003 *
     2.1888 2.4266 0.1669
```

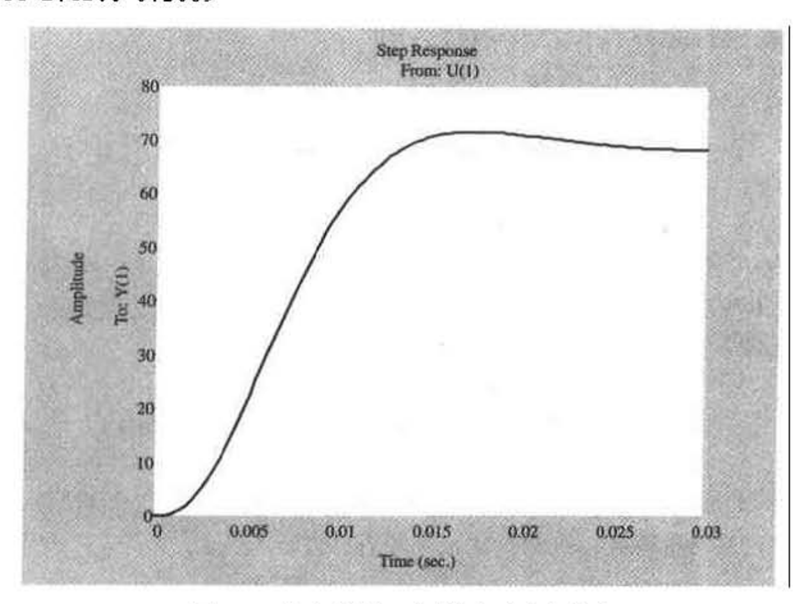


图 8-5　状态反馈闭环系统阶跃响应曲线

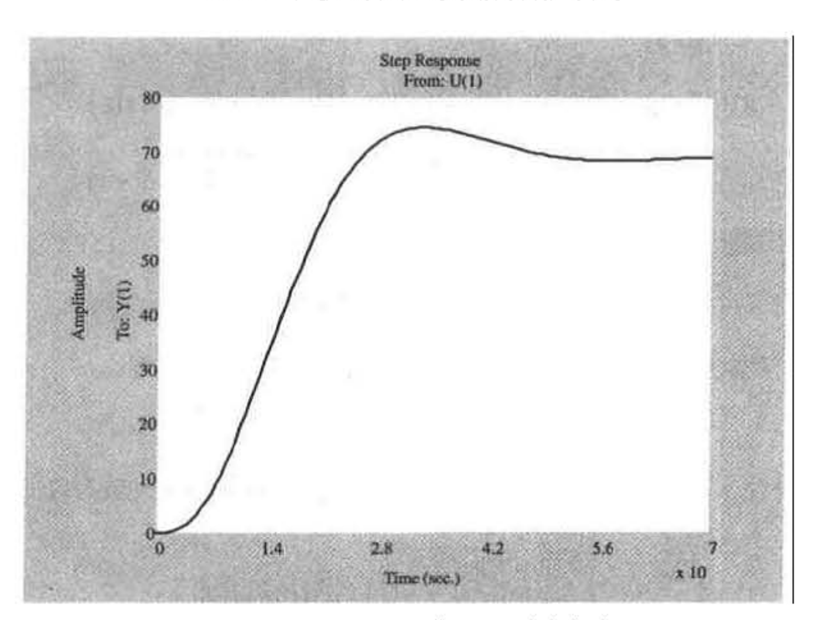


图 8-6　输出反馈闭环系统阶跃响应曲线

对比两条曲线可以看出，采用输出反馈的闭环系统单位阶跃给定响应曲线比采用状态反馈的超调量稍大，且有一次振荡，峰值时间也短（约 3.2×10^{-3} 秒）。采用状态反馈的超调量较小，并且超调后单调衰减。这确实是状态反馈的为最优控制，输出反馈的为次优控制。

8.3 离散系统稳态线性二次型最优控制的 MATLAB 实现

8.3.1 离散系统稳态线性二次型最优控制

设完全可控线性离散系统的状态方程为：

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}\boldsymbol{u}(k),\quad \boldsymbol{x}(0)=\boldsymbol{x}_0 \tag{8-24}$$

$$(k=0,1,\cdots,N-1)$$

式中，$\boldsymbol{x}(k)$为 n 维状态向量；$\boldsymbol{u}(k)$为 r 维控制向量，且不受约束；$\boldsymbol{A}$ 为 $n\times n$ 维非奇异矩阵，$\boldsymbol{B}$ 为 $n\times r$ 维矩阵。

系统的性能指标为：

$$J=\frac{1}{2}\boldsymbol{x}^T(N)\boldsymbol{S}\boldsymbol{x}(N)+\frac{1}{2}\sum_{k=0}^{N-1}\left[\boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k)\right] \tag{8-25}$$

式中，$\boldsymbol{Q}$ 为 $n\times n$ 维正定或半正定实对称矩阵；$\boldsymbol{R}$ 为 $r\times r$ 维正定实对称矩阵；$\boldsymbol{S}$ 为 $n\times n$ 维正定或半正定实对称矩阵。则有最优反馈矩阵（之一）为：

$$\boldsymbol{K}(k)=\left[\boldsymbol{R}+\boldsymbol{B}^T\boldsymbol{P}(k+1)\boldsymbol{B}\right]^{-1}\boldsymbol{B}^T\boldsymbol{P}(k+1)\boldsymbol{A} \tag{8-26}$$

与之对应的最优控制序列：

$$\boldsymbol{u}^*(k)=-\boldsymbol{K}(k)\boldsymbol{x}(k)=-\left[\boldsymbol{R}+\boldsymbol{B}^T\boldsymbol{P}(k+1)\boldsymbol{B}\right]^{-1}\boldsymbol{B}^T\boldsymbol{P}(k+1)\boldsymbol{A}\boldsymbol{x}(k) \tag{8-27}$$

$$(k=0,1,\cdots,N-1)$$

以及最优性能指标：

$$J^*=\frac{1}{2}\boldsymbol{x}^T(0)\boldsymbol{P}(0)\boldsymbol{x}(0) \tag{8-28}$$

以上几式中，$\boldsymbol{P}(k)$为正定矩阵，它是以下黎卡提差分方程的正定解：

$$\boldsymbol{P}(k)=\boldsymbol{Q}+\boldsymbol{A}^T\left[\boldsymbol{P}^{-1}(k+1)+\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\right]^{-1}\boldsymbol{A} \tag{8-29}$$

若控制步数 N 为无限值，即令 $N\to\infty$，系统最优控制的解成为稳态解。系统性能指标则改为：

$$J=\frac{1}{2}\sum_{k=0}^{\infty}\left[\boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k)\right] \tag{8-30}$$

$\boldsymbol{K}(k)$变成常数增益矩阵 $\boldsymbol{K}$：

$$\boldsymbol{K}=\left[\boldsymbol{R}+\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{B}\right]^{-1}\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{A} \tag{8-31}$$

$\boldsymbol{P}(k)$也变成常数矩阵 $\boldsymbol{P}$：

$$P=Q+A^T\left[P^{-1}+BR^{-1}B^T\right]^{-1}A \tag{8-32}$$

对应的最优控制序列为：

$$u^*(k)=-K\,x(k)=-\left[R+B^TPB\right]^{-1}B^TPAx(k) \tag{8-33}$$

闭环系统的状态方程为：

$$x(k+1)=Ax(k)+Bu(k)=\left[A-B\left(R+B^TPB\right)^{-1}B^TPA\right]x(k)=\left(I+BR^{-1}B^TP\right)^{-1}Ax(k) \tag{8-34}$$

以及最优性能指标仍为：

$$J^*=\frac{1}{2}x^T(0)P(0)x(0) \tag{8-35}$$

8.3.2 离散系统稳态线性二次型最优控制的 MATLAB 实现

在 MATLAB 的工具箱里，有特别提供的函数来求解离散系统稳态线性二次型状态调节器问题。其函数有 dlqr ( ) 与 dlqry ( )。函数的调用格式为：

[*K*,*S*,*E*]=dlqr(*A*,*B*,*Q*,*R*,*N*)

[*K*,*S*,*E*]=dlqry(*A*,*B*,*C*,*D*,*Q*,*R*,*N*)

其中，输入参量 ***A*** 为系统的状态矩阵；***B*** 为系统的输入矩阵；***Q*** 为给定的正定或半正定实对称矩阵；***R*** 为给定的正定实对称矩阵；***N*** 代表更一般化性能指标中交叉乘积项的加权矩阵。输出参量 ***K*** 为离散最优反馈增益矩阵；***S*** 为对应 Riccati 方程的唯一正定解 ***P***；***E*** 为 ***A-BK*** 的特征值。

函数 dlqry ( ) 用来求解二次型状态调节器的特例，这个特例就是用输出反馈替代状态反馈，即有：

$$u(k)=-K\,y(k) \tag{8-36}$$

其性能指标则为：

$$J=\frac{1}{2}\sum_{k=0}^{\infty}\left[y^T(k)Qy(k)+u^T(k)Ru(k)\right] \tag{8-37}$$

【例 8-4】已知伺服系统动态结构图如图 8-7 所示：

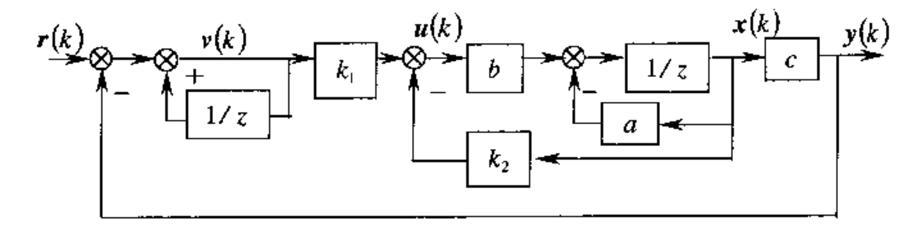


图 8-7 伺服系统动态结构图

且

$$x(k+1)=ax(k)+bu(k)$$

$$\boldsymbol{y}(k)=\boldsymbol{c}\boldsymbol{x}(k)$$

式中

$$a=0.5,\quad b=1,\quad c=1,\quad d=0$$

试计算稳态最优反馈增益矩阵并求系统闭环后的单位阶跃给定响应。

【解】

由结构图有：

$$\boldsymbol{x}(k+1)=\boldsymbol{a}\boldsymbol{x}(k)+\boldsymbol{b}\boldsymbol{u}(k)$$

$$\boldsymbol{u}(k)=\boldsymbol{k}_1\boldsymbol{v}(k)-\boldsymbol{k}_2\boldsymbol{x}(k)$$

$$\boldsymbol{v}(k)=\boldsymbol{r}(k)-\boldsymbol{y}(k)+\boldsymbol{v}(k-1)$$

$$\boldsymbol{y}(k)=\boldsymbol{c}\boldsymbol{x}(k)$$

由$\boldsymbol{v}(k)$有：

$$\boldsymbol{v}(k+1)=\boldsymbol{r}(k+1)-\boldsymbol{y}(k+1)+\boldsymbol{v}(k)=-\boldsymbol{c}\boldsymbol{a}\boldsymbol{x}(k)+\boldsymbol{v}(k)-\boldsymbol{c}\boldsymbol{b}\boldsymbol{u}(k)+\boldsymbol{r}(k+1)$$

由$\boldsymbol{x}(k+1)$与$\boldsymbol{v}(k+1)$写出矩阵：

$$\begin{bmatrix}\boldsymbol{x}(k+1)\\\boldsymbol{v}(k+1)\end{bmatrix}=\begin{bmatrix}a & 0\\-ca & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k)\\\boldsymbol{v}(k)\end{bmatrix}+\begin{bmatrix}b\\-cb\end{bmatrix}\boldsymbol{u}(k)+\begin{bmatrix}0\\1\end{bmatrix}\boldsymbol{r}(k+1)\qquad(8\text{-}38)$$

对于系统稳态，$k=\infty$，以上矩阵就成为：

$$\begin{bmatrix}\boldsymbol{x}(\infty)\\\boldsymbol{v}(\infty)\end{bmatrix}=\begin{bmatrix}a & 0\\-ca & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(\infty)\\\boldsymbol{v}(\infty)\end{bmatrix}+\begin{bmatrix}b\\-cb\end{bmatrix}\boldsymbol{u}(\infty)+\begin{bmatrix}0\\1\end{bmatrix}\boldsymbol{r}(\infty)$$

令$\boldsymbol{x}_e(k)=\boldsymbol{x}(k)-\boldsymbol{x}(\infty)$，$\boldsymbol{v}_e(k)=\boldsymbol{v}(k)-\boldsymbol{v}(\infty)$，$\boldsymbol{u}_e(k)=\boldsymbol{u}(k)-\boldsymbol{u}(\infty)$。将以上两矩阵相减，可得：

$$\begin{bmatrix}\boldsymbol{x}_e(k+1)\\\boldsymbol{v}_e(k+1)\end{bmatrix}=\begin{bmatrix}a & 0\\-ca & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_e(k)\\\boldsymbol{v}_e(k)\end{bmatrix}+\begin{bmatrix}b\\-cb\end{bmatrix}\boldsymbol{u}_e(k)$$

由$\boldsymbol{u}(k)=\boldsymbol{k}_1\boldsymbol{v}(k)-\boldsymbol{k}_2\boldsymbol{x}(k)$有

$$\boldsymbol{u}_e(k)=\boldsymbol{k}_1\boldsymbol{v}_e(k)-\boldsymbol{k}_2\boldsymbol{x}_e(k)$$

再令$\boldsymbol{x}_1(k)=\boldsymbol{x}_e(k)$，$\boldsymbol{x}_2(k)=\boldsymbol{v}_e(k)$，$\boldsymbol{w}(k)=\boldsymbol{u}_e(k)$。以上矩阵可写成：

$$\begin{bmatrix}\boldsymbol{x}_1(k+1)\\\boldsymbol{x}_2(k+1)\end{bmatrix}=\begin{bmatrix}a & 0\\-ca & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1(k)\\\boldsymbol{x}_2(k)\end{bmatrix}+\begin{bmatrix}b\\-cb\end{bmatrix}\boldsymbol{w}(k)\qquad(8\text{-}39)$$

式中

$$w(k)=-\begin{bmatrix}k_2 & -k_1\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix} \tag{8-40}$$

当系统有单位阶跃给定输入时，$r(k+1)=r(\infty)=r$。将$u(k)=k_1 v(k)-k_2 x(k)$代入式（8-38）矩阵可得：

$$\begin{bmatrix}x(k+1)\\v(k+1)\end{bmatrix}=\begin{bmatrix}a & 0\\-ca & 1\end{bmatrix}\begin{bmatrix}x(k)\\v(k)\end{bmatrix}+\begin{bmatrix}b\\-cb\end{bmatrix}\begin{bmatrix}-k_2 & k_1\end{bmatrix}\begin{bmatrix}x(k)\\v(k)\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}r$$

根据矩阵乘法运算规则，整理以上矩阵有：

$$\begin{bmatrix}x(k+1)\\v(k+1)\end{bmatrix}=\begin{bmatrix}a-bk_2 & bk_1\\-ca+cbk_2 & 1-cbk_1\end{bmatrix}\begin{bmatrix}x(k)\\v(k)\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}r \tag{8-41}$$

还有：

$$y(k)=cx(k)=\begin{bmatrix}c & 0\end{bmatrix}\begin{bmatrix}x(k)\\v(k)\end{bmatrix} \tag{8-42}$$

设定性能指标为：

$$J=\frac{1}{2}\sum_{k=0}^{\infty}\left[x^T(k)Qx(k)+w^T(k)Rw(k)\right] \tag{8-43}$$

式中参量Q、R选择为：

$$Q=\begin{bmatrix}1000 & 0\\0 & 1\end{bmatrix},\quad R=[1]$$

根据以上分析以及式（8-38）、（8-41）与（8-42），为计算系统最优反馈增益矩阵并求系统闭环后的阶跃响应，特给出程序 L8301.m 如下。

```
% MATLAB PROGRAM L8301.m
%
a=0.5;b=1;c=1;d=0;
Q=[1000 0;0 1];R=[1];
A=[a 0;-c*a 1];
B=[b;-c*b];
Kx=dlqr(A,B,Q,R)
k1=-Kx(2);k2=Kx(1);
axc=[(a-b*k2) b*k1;(-c*a+c*b*k2) (1-c*b*k1)]
bxc=[0;1];cxc=[1 0];dxc=[0];
dstep(axc,bxc,cxc,dxc,1,100)
```

程序运行后可得系统最优状态反馈增益矩阵 ***Kx***

```
Kx = 0.4995   -0.0311
```

程序运行后还绘制出系统闭环后的单位阶跃给定响应曲线如图 8-8 所示。

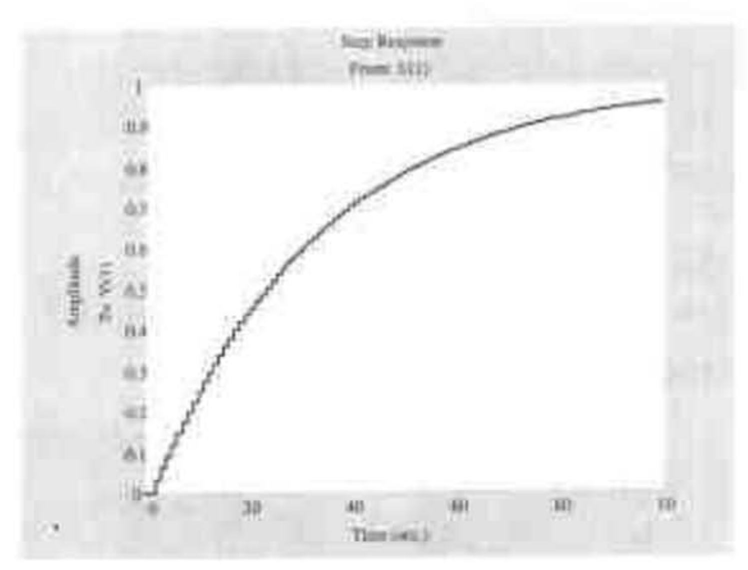


图 8-8 闭环系统单位阶跃给定响应曲线

【例 8-5】已知如图 8-4 所示的可控整流装置供电给直流电机传动系统。试对系统按 T=0.1s 设计数字最优控制系统（计算稳态最优反馈增益矩阵）并求闭环离散系统的阶跃给定响应。

【解】

首先调用函数 linmod2 ( ) 将结构图模型转换为状态空间模型：

$$\dot{x}(t)=a_1x(t)+b_1u(t)$$
$$y(t)=c_1x(t)+d_1u(t)$$

对系统按 T=0.1 s 进行采样离散化，需调用将连续系统离散化的 MATLAB 函数 c2d ( )，这个函数在 7.2 节里已经介绍过。

可得离散系统模型：

$$x(k+1)=ax(k)+bu(k)$$
$$y(k)=cx(k)+du(k)$$

于是有数字最优控制系统仍如图 8-7 所示。因此有以上关系式（8-39）、（8-40）与（8-43）。即性能指标为：

$$J=\frac{1}{2}\sum_{k=0}^{\infty}\left[x^T(k)Qx(k)+w^T(k)Rw(k)\right]$$

式中选择 Q、R 为：

$$Q=\begin{bmatrix}1000 & 0\\ 0 & 1\end{bmatrix},\quad R=[1]$$

根据以上分析，为计算稳态最优状态反馈增益矩阵并绘制系统闭环后的阶跃给定响应曲线，特给出程序 L8302.m 如下。

```
% MATLAB PROGRAM L8302.m
%
[a1,b1,c1,d1]=linmod2('t8204');
Q=diag([10 1 10 1]);R=[1];
```

```
[a,b]=c2d(a1,b1,0.1);c=c1;d=d1;
A=[a zeros(3,1);-c*a 1];
B=[b;-c*b];Kx=dlqr(A,B,Q,R)
k1=-Kx(2);k2=Kx(1);
axc=A-B*Kx;
bxc=[0;0;0;1];cxc=[0 1 0 0];dxc=[0];
dstep(axc,bxc,cxc,dxc,1,100)
```

程序运行后可得系统最优反馈增益矩阵 ***Kx*** 与绘制系统闭环后的单位阶跃给定响应曲线如图 8-9 所示。

```
Kx = 0.1036    6.5696   0.1451   -0.0059
```

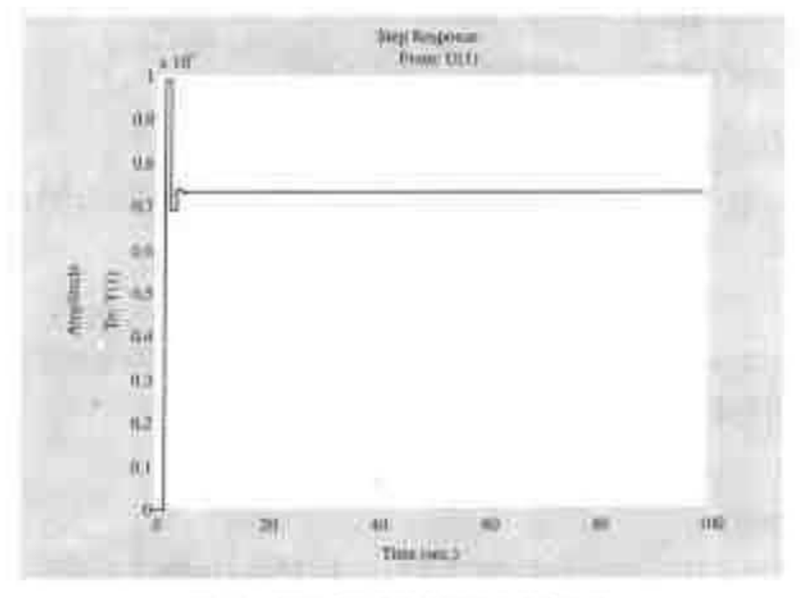


图 8-9 闭环系统单位阶跃给定响应曲线

8.4 最优观测器的 MATLAB 实现

系统的 Kalman 滤波器就是最优观测器。对于带有系统噪声与量测噪声的实际系统，必须通过适当的结构，抑制或滤掉噪声对系统的干扰及影响，对系统的状态做出充分精确的估计。利用 Kalman 滤波器对系统进行最优控制是非常有效的。

8.4.1 连续时不变系统的 Kalman 滤波

给定系统的状态方程与量测方程分别为：

$$\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)+\boldsymbol{G}\boldsymbol{w}(t) \tag{8-44}$$

$$\boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)+\boldsymbol{v}(t) \tag{8-45}$$

式中，$\boldsymbol{x}(t)$为 n 维状态向量；$\boldsymbol{u}(t)$为 r 维控制向量；$\boldsymbol{y}(t)$为 m 维量测向量；$\boldsymbol{A}$ 为 $n \times n$ 维常数矩阵；$\boldsymbol{B}$ 为 $n \times r$ 维常数矩阵；$\boldsymbol{G}$ 为 $n \times r$ 维常数矩阵；$\boldsymbol{C}$ 为维 $m \times n$ 常数矩阵；$\boldsymbol{D}$ 为 $m \times r$ 维常数矩阵；假定 $\boldsymbol{w}(t)$ 为随机噪声干扰输入，它是零均值的 r 维白噪声过程；假定 $\boldsymbol{v}(t)$ 为随机量测噪声，是零均值的 m 维白噪声过程。$\boldsymbol{w}(t)$与 $\boldsymbol{v}(t)$两噪声过程均平稳且互不相关。即有：

$E\{\boldsymbol{w}(t)\}=0$，对一切 $t \geqslant t_0$

$E\{\boldsymbol{v}(t)\}=0$，对一切 $t \geqslant t_0$

$E\{\boldsymbol{w}(t)\boldsymbol{w}^T(\tau)\}=\boldsymbol{Q}_0\delta(t-\tau)$，对一切 t，$\tau \geqslant t_0$

此式中的 $\boldsymbol{Q}_0$ 为常数矩阵（叫做模型噪声的协方差矩阵）。

$E\{\boldsymbol{v}(t)\boldsymbol{w}^T(\tau)\}=\boldsymbol{Q}_0\delta(t-\tau)$，对一切 t，$\tau \geqslant t_0$

此式中的 $\boldsymbol{R}_0$ 为常数矩阵（叫做量测噪声的协方差矩阵）。

$E\{\boldsymbol{w}(t)\boldsymbol{v}^T(\tau)\}=0$，对一切 t，$\tau \geqslant t_0$

令 $\hat{\boldsymbol{x}}(t)$与 $\tilde{\boldsymbol{x}}(t)$分别为状态向量估计值与状态向量的估计误差值，$\boldsymbol{x}(t)$为状态向量的理论值，则有：

$$\tilde{\boldsymbol{x}}(t)=\boldsymbol{x}(t)-\hat{\boldsymbol{x}}(t) \tag{8-46}$$

除上述假定外，还假定$\{\boldsymbol{C}, \ \boldsymbol{A}\}$是完全可观测的。在这些假定均成立的条件下，使估计误差平方和的期望值最小（最小方差迹准则滤波估计）即有：

$$J=E\left[\tilde{\boldsymbol{x}}^T(t)\tilde{\boldsymbol{x}}(t)\right]=\min \tag{8-47}$$

其最优估计器为：

$$\dot{\hat{\boldsymbol{x}}}(t)=\boldsymbol{A}\hat{\boldsymbol{x}}(t)+\boldsymbol{B}\boldsymbol{u}(t)+\boldsymbol{L}\left[\boldsymbol{y}(t)-\boldsymbol{C}\hat{\boldsymbol{x}}(t)\right]=(\boldsymbol{A}-\boldsymbol{L}\boldsymbol{C})\hat{\boldsymbol{x}}(t)+\boldsymbol{B}\boldsymbol{u}(t)+\boldsymbol{L}\boldsymbol{y}(t) \tag{8-48}$$

式中

$$\boldsymbol{L}=\boldsymbol{P}_0\boldsymbol{C}^T\boldsymbol{R}_0^{-1} \tag{8-49}$$

其中 $\boldsymbol{P}_0$ 为以下 Riccati 方程的解：

$$\boldsymbol{A}\boldsymbol{P}_0+\boldsymbol{P}_0\boldsymbol{A}^T+\boldsymbol{G}\boldsymbol{Q}_0\boldsymbol{G}^T-\boldsymbol{P}_0\boldsymbol{C}^T\boldsymbol{R}_0^{-1}\boldsymbol{C}\boldsymbol{P}_0=0 \tag{8-50}$$

可以证明，Riccati 方程的解 $\boldsymbol{P}_0$ 就是估计误差的协方差，而此协方差的迹（$tr\,\boldsymbol{P}_0$）即为误差方差。因此有：

$$tr\,\boldsymbol{P}_0=tr\,E\left[\tilde{\boldsymbol{x}}(t)\tilde{\boldsymbol{x}}^T(t)\right]=E\left[\tilde{\boldsymbol{x}}^T(t)\tilde{\boldsymbol{x}}(t)\right] \tag{8-51}$$

8.4.2 Kalman 滤波的 MATLAB 实现

在 MATLAB 的工具箱里提供了 kalman ( ) 函数来求解系统的 Kalman 滤波器。函数的调用格式为：

[*Kest*,***L***,***P***] = **kalman**(*sys*,***Q***,***R***,***N***)

其中，输入参量 *sys* 为连续或离散系统带扰动的状态空间模型（$\boldsymbol{A}, \tilde{\boldsymbol{B}}, \boldsymbol{C}, \tilde{\boldsymbol{D}}$），当模型有两个输入时，$\tilde{\boldsymbol{B}}$=[$\boldsymbol{B}$，$\boldsymbol{G}$]，$\tilde{\boldsymbol{D}}$=[$\boldsymbol{D}$，$\boldsymbol{D}$]；$\boldsymbol{Q}$ 为模型噪声的协方差矩阵；$\boldsymbol{R}$ 为量测噪声的协

方差矩阵；N为可选项，它对应模型噪声与量测噪声的相关项。输出参量 Kest 为 Kalman 滤波器的状态估计器，其动态方程如下：

$$\dot{\hat{x}}(t)=(A-LC)\hat{x}(t)+(B-LD)u(t)+Ly(t) \tag{8-52}$$

$$\begin{bmatrix}\hat{y}(t)\\ \hat{x}(t)\end{bmatrix}=\begin{bmatrix}C\\ I\end{bmatrix}\hat{x}(t)+\begin{bmatrix}D\\ 0\end{bmatrix}u(t) \tag{8-53}$$

L为 Kalman 滤波器的增益矩阵；P为对应的 Riccati 方程的解，即估计误差的协方差。

在 MATLAB 老版本的工具箱里还提供了 lqe ( ) 函数来求解系统的 Kalman 滤波器。函数的调用格式为：

[L,P,E] = lqe(A,G,C,Q,R,N)

其中，输入参量 A、G、C为系统式（8-44）与（8-45）中的对应参量；Q为模型噪声的协方差矩阵；R为量测噪声的协方差矩阵；N为可选项，它对应模型噪声与量测噪声的相关项。输出参量L为 Kalman 滤波器的增益巨阵；P为对应的 Riccati 方程的解；E为估计器的闭环特征值。

【例 8-6】已知系统的状态方程与量测方程分别为：

$$\dot{x}(t)=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}x(t)+\begin{bmatrix}0\\ 1\end{bmatrix}u(t)+\begin{bmatrix}1\\ 1\end{bmatrix}w(t)$$

$$y(t)=\begin{bmatrix}1 & 0\end{bmatrix}x(t)+v(t)$$

其中，$w(t)$与$v(t)$都是零均值的白噪声，且$w(t)$=1e-1，$v(t)$=1e-3。试计算系统 Kalman 滤波器的增益矩阵与估计误差的协方差。

【解】

为计算系统 Kalman 滤波器的增益矩阵与估计误差的协方差，给出以下程序 L8401.m。

```
% MATLAB PROGRAM L 8401.m
%
a0=[0 1;-1 0];b0=[0;1];c0=[1 0];d0=0;
s1=ss(a0,b0,c0,d0);
q0=1e-1;r0=1e-3;
[Kest,L,P]=kalman(s1,q0,r0);L,P
```

程序运行后，可得系统 Kalman 滤波器的增益矩阵 L 与估计误差的协方差 P 分别为：

```
L =
    4.2544
    9.0499
P =
    0.0043   0.0090
    0.0090   0.0428
```

【例 8-7】已知可控直流电源供电给直流电机的系统结构图如图 8-4 所示。试对系统进行 Kalman 滤波器的设计并对经 Kalman 滤波后的系统闭环进行阶跃响应仿真。

【解】

根据题意，要对系统进行 Kalman 滤波器的设计，给出调用函数 kalman()与 linmod()的程序 L8402.m。

```
% MATLAB PROGRAM L8402.m
%
[a,b,c,d]=linmod2('t8204');s0=ss(a,b,c,d);
f=[1;0;0];q=[1 0 0;0 1 0;0 0 1];r=1;q0=1;r0=1;
K=lqr(a,b,q,r);
s1=ss(s0.a,[s0.b,s0.b],s0.c,[s0.d,s0.d]);
[kest,L,P]=kalman(s1,q0,r0);L,P
a1=s0.a-s0.b*K-L*s0.c;
sys=s0*ss(a1,L,K,0);sysc=feedback(sys,1);
t=0:0.01:1;step(sysc,t)
```

程序运行后，可得到 Kalman 滤波器的增益矩阵 L 与估计误差的协方差 P 分别为：

```
L =
    7.3619
    0.1031
   31.9083
P =
    0.1063    0.0015    0.4607
    0.0015    0.0008    0.0281
    0.4607    0.0281    3.8113
```

程序运行后还得到经 Kalman 滤波后的系统闭环单位阶跃给定响应仿真曲线（如图 8-10 所示）。

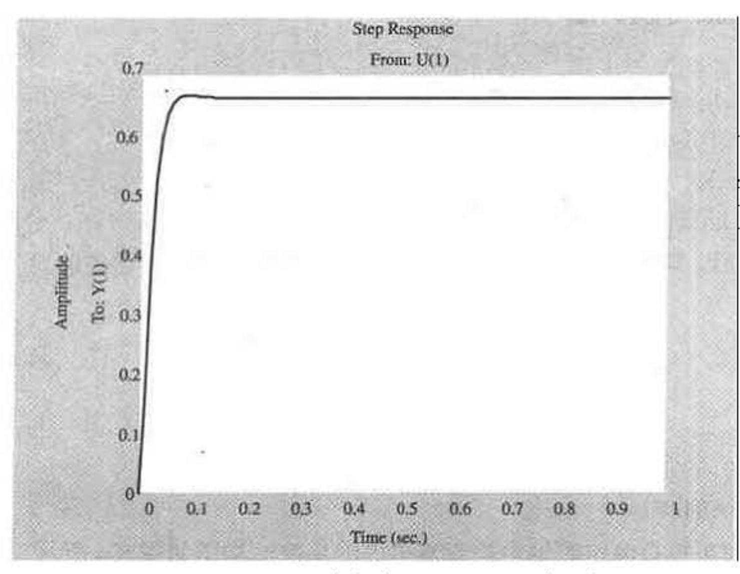


图 8-10　Kalman 滤波系统阶跃响应仿真曲线

由图 8-10 可见，经 Kalman 滤波后，系统闭环的单位阶跃给定响应是非常理想的，仿真曲线为几乎立即响应的阶跃方波，只稍微有一点点超调，曲线就趋于水平。

8.5 线性二次型 Guass 最优控制的 MATLAB 实现

考虑系统随机输入噪声与随机量测澡声的线性二次型最优控制叫做线性二次型高斯（Guass）最优控制，即 LQG 控制。线性二次型高斯最优控制是输出反馈控制。这种计及系统受到随机因素的作用而采取的控制策略，对解决线性二次型最优控制问题，显然更具有实用性。

8.5.1 LQG 最优控制的求解

给定系统的状态方程与量测方程分别为：

$$\dot{x}(t)=Ax(t)+Bu(t)+Gw(t) \tag{8-54}$$

$$y(t)=Cx(t)+Du(t)+v(t) \tag{8-55}$$

式中，参量 $x(t)$、$u(t)$、$y(t)$、A、B、G、C、D 含义同 8.4 节。假定 $w(t)$ 为随机噪声干扰输入，它是零均值的 r 维白噪声过程；假定 v（t）为随机量测噪声，是零均值的 m 维白噪声过程。$w(t)$ 与 v（t）两噪声过程均平稳且互不相关。系统的性能指标为：

$$J=E\left\{\int_0^{\infty}\left[x^TQx+u^TRu\right]dt\right\} \tag{8-56}$$

根据 LQG 问题的分离原理，LQG 最优控制是两个方面问题的综合：一是二次型调节器问题，二是最优估计器问题。

（1）LQ 最优控制就是二次型调节器问题。在 8.2 节已经介绍过，最优状态反馈控制有最优反馈矩阵式(8-17)，最优控制为式(8-18)，以及 P 满足的 Riccati 方程式(8-20)。

（2）利用 kalman 滤波理论，从状态 $x(t)$ 中得到最优估计 $\hat{x}(t)$，使估计误差平方和的期望值最小（最小方差迹准则滤波估计），即有式 (8-47)，这种最优估计器为式(8-48)，其中，L 为 Kalman 滤波器增益，即式(8-49)，其中 P_0 为 Riccati 方程式（8-50）的解。

分别计算 LQ 最优控制与最优估计，尔后将这两个问题的解综合在一起，就得到 LQG 控制的最优解。

8.5.2 LQG 最优控制的 MATLAB 实现

在 MATLAB 的工具箱里，有特别提供的函数 reg ( ) 来求解 LQG 最优控制。函数的调用格式为：

rsys = reg(*sys*,*K*,*L*)

其中，输入参量 *sys* 为系统的状态空间模型；***K*** 为由函数 lqr( ) 求得的最优状态反馈增益矩阵；***L*** 为函数 lqe ( ) 求得的 kalman 滤波器状态估计增益矩阵。

【例 8-8】已知可控直流电源供电给直流电机的系统结构图如图 8-4 所示。试对系统进行 LQG 最优控制。

【解】

根据题意，要对系统进行 LQG 最优控制，给出调用函数 lqr()、lqe()与 reg()的程序 L8501.m。

```
% MATLAB PROGRAM L8501.m
%
[a,b,c,d]=linmod2('t8204');s1=ss(a,b,c,d);
f=[1;0;0];q=[1 0 0;0 1 0;0 0 1];r=1;q0=1;r0=1;
K=lqr(a,b,q,r);L=lqe(a,f,c,q0,r0);
[af,bf,cf,df]=reg(a,b,c,d,K,L);
sf=ss(af,bf,cf,df);
sys=feedback(s1,sf);
t=0:0.01:1;step(sys,t)
```

程序运行后得到进行 LQG 最优控制系统闭环的单位阶跃给定响应仿真曲线（如图 8-11 所示）。

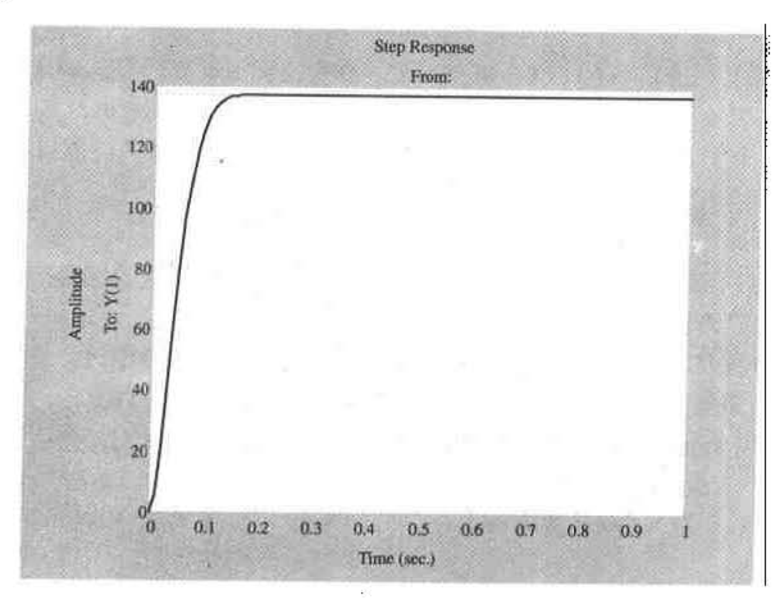


图 8-11　LQG 最优控制的系统阶跃响应仿真曲线

由图 8-11 可见，LQG 最优控制的系统与经 kalman 滤波的系统，两者闭环后其单位阶跃给定响应基本一致，仿真曲线为几乎是立即响应的阶跃方波，确实非常理想。

附录 A　MATLAB 常用函数

MATLAB 程序设计与控制系统计算、仿真中经常使用的函数（不包括工具箱函数）计有：数据分析函数、基本数学函数、专用数学函数、基本矩阵与数组运算函数、矩阵函数、多项式与插值函数等。现将这些 MATLAB 基本函数命令按英语字母顺序索引如下，供读者查询。关于函数的使用格式，请参见联机帮助。

函 数 名	函 数 功 能
abs	取决对值，或将字符转换为 ASCII 值向量
acopy	苹果机上把文件从一个文件夹拷入另一个文件夹
acos	反余弦函数
acos	反余弦
acosh	反双曲余弦函数
acot	反余切函数
acoth	反双曲余切函数
acsc	反余割函数
acsch	反双曲余割函数
addpath	添加 MATLAB 的搜索路径
airfoil	nasa 翼面稀疏矩阵显示
airy	计算第一类与第二类 Airy 函数
all	向量元素逻辑与
allvalues	求 RootOf 所确定表达式的值
amove	苹果机上把文件从一个文件夹移入另一个文件夹
angle	相角函数
ans	最近生成的无名结果
any	测试向量中是否有为真元素
applescript	苹果机上从一个文件中载入编译后的文件并执行
area	区域填充
arename	苹果机上文件重命令
areveal	在苹果机字桌面上显示文件名
argnames	返回在线函数对象的输入参数
arith	matlab 的各种算术运算符信息
asec	反正割函数
asech	反双曲正割
asin	反正弦函数
asinh	反双曲正弦
atan	反正切函数
atan2	四象限反正切
atanh	反双曲正切
auad8	Newton-Cotes 数值积分

（续）

auread	读 au 声音文件
autumn	由红色和黄色生成的色图矩阵
auwrite	写 au 声音文件
axes	创建轴对象
axis	设置坐标刻度范围
balance	提高特征值精度的选项
bandemo	极小化香蕉函数的演示
bar	条形图
bar3	垂直三维条形图
bar3h	水平三维条形图
barh	水平二维条形图
base2dec	任意进制数化为十进制数
bench	matlab 测试基准问题
besselh	第三类 Bessel 函数
besseli、besselk	修正 Bessel 函数
besselj、bessely	Bessel 函数
beta	Beta 函数
betainc、betaln	Beta 函数
bicg	双共轭梯度法
bicgstab	双共轭梯度稳定法
bighten	改变色图矩阵亮度
bin2dec	把二进制转换为十进制
bitand	位和
bitcmp	补码
bitget	按位取补
bitmax	机器的最大浮整数
bitor	位异或
bitset	给位赋值
bitshift	移位
bitxor	位异或
blanks	设置一个由空格组成的字符串
bone	带有蓝色的灰度颜色表
break	中断循环语句的执行
brighten	使图形色调变亮
bucky	buckminister fuller 拱形演示
builtin	用超载方法运行内部函数
calendar	日历
cart2pol	把直角坐标转换为极坐标或圆柱坐标
cart2sph	把直角坐标转换为球坐标
case	swith 结构关键字
cat	拼接多维数组
caxis	伪颜色坐标轴设定
cd	改变当前工作目录
cdf2rd	复块对角矩阵到实块对角阵转换
cdf2rdf	复数对角型转换为实块对角型
ceedit	设置命令行编辑与回调的参数
ceil	朝正无穷方向取整

（续）

cell	生成细胞数组
cell2struct	把细胞数组转换为结构数组
celldisp	显示单元数组的单元
cellplot	用图形方式图示细胞数组的结构
cellstr	用字符数组生成细胞数组
census	2000 年美国人口普查预测
cgs	平方共轭剃度法
char	将 Cell 数组转化为字符串数组，将函数对象转化为生成函数的命令，将数值转化字符
charploy	符号特征多项式
choices	生成按六式演示菜单
chol	Cholesky 分解
cholinc	不完全 Cholesky 分解
cla	清除当前坐标轴
clabel	等值线加注高度标记
class	返回指定对象的类型或建立指定类型的对象
clc	清除命令窗口
clear	从内存中删除变量和函数
clf	清除当前图形窗口
click	当前日期和时间向量`
clock	当前时间函数
clommd	最小列的阶次
close	关闭图形窗口
collet	合并系数
colmmd	列最小度排序
colon	冒号表达式的帮助信息
colorbar	色图矩阵的色条
colorcube	规则的色图矩阵
colormap	设定颜色可查表
colormenu	颜色表演示
colperm	由非零数据的计数来排列各列
colspace	列空间的基
comapn	伴随矩阵
comet	彗星状轨迹绘制
comet3	绘制三维彗星状的轨迹
compan	生成伴随矩阵
compass	复数向量图
compose	复合函数
computer	计算机类型测试
cond	和逆相关的条件数
condeig	和特征值相关的条件数
condest	估算范数
conj	共轭复数函数
constr	求解约束最优化问题
contour	二元函数等值线
contour3	二元函数三维等值线
conv	求两多项式的卷积

（续）

conv2	二维卷积
convhull	凸壳函数
cool	天蓝粉色基色颜色表
copper	线性铜色调颜色表
corrcoef	相关函数系数
cos	余弦函数
cosh	双曲余弦
cot	余切函数
coth	双曲余切
cov	协方差矩阵
cplxdemo	复变量函数映射函数演示
cplxpair	将数据按共轭复数对重新排序
cputime	所用的 cpu 时时间
cross	向量外积
csc	余割函数
csch	双曲余割
cumdum	累计和
cumprod	累计积
cumsum	累计和
cumtrapz	累计梯形积分
cusum	各元素累加积
cylinder	产生柱体
date	当前日期字符串
datenum	日期串转换为日期号
datestr	日期号转换为日期串
datetick	日期串为坐标刻度
datevec	日期串转换为向量
dbclear	清除跟踪调试断点
dbcont	恢复程序运行
dbdown	改变工作空间到下级
dblquad	二维数值积分
dbmex	调试 MEX 文件
dbquit	退出调试模式
dbstack	显示将执行的命令序列
dbstatus	列出所有断点
dbstep	从断点处继续运行一行或多行
dbstop	设置断点
dbtype	列出 M 文件，每行带行号
dbup	改变局部工作空间内容
deal	访问单元数组的元素
deblank	去掉字符串末尾的空格
dec2base	将十进制数转化为任意进制数
dec2bin	将十进制数转化为二进制数
dec2hex	十进制数转化为 16 进制数
dec9onv	两多项式相除
deconv	因式分解与多项式除法
del2	五点式离散 laplace 变换

（续）

delaunay	Delaunay 三角化
delete	删除图形对象、文本文件
delsq	Laplace 方程 5 点差分格式矩阵
delsqdemo	各种域上的有限差分演示
demo	运行 matlab 演示程序
det	求矩阵的行列式
determ	符号行列式
diag	生成对角矩阵或取出对角元素
diary	将 matlab 运行的命令存盘
diff	差分和近似微分
diffuse	图象柔焦处理
digits	设置精度位数
dir	列出当前目录的内容
disp	显示矩阵和数组内容
dlmread	把 ASCII 中的数据读入矩阵
dlmwrite	把矩阵写入 ASCII 文件
dmperm	dulmage-mendelsohn 分解
doc	启动超文本帮助
double	转换为双精度型
drawnow	闪烁未定的图象文件
dsolve	常微分方程符号解
earthmap	地球拓扑图形的显示
echo	显示 M 文件执行时是否显示命令的开关
edit	编辑一个 M 文件
eig	求矩阵的特征值与特征向量
eigensus	符号矩阵特征系统
eigmovie	对称矩阵特征值求解过程演示
eigs	求少数特征值和特征向量
eiplot	一元函数曲线绘图器
ellipj	椭圆 Jacobi 函数
ellipke	第一二类完全椭圆积分
else	与 if 一起使用的转移语句
elseif	与 IF 一起使用的转移语句
end	for while switch 和 if 结构的关键字或者下标的最后一个
eomday	某年月的最后一天
eps	浮点数相对误差
erf、erfc	Error 函数
erfcx、erfinv	Error 函数
error	显示错误信息并中断函数
errorbar	误差条形图绘制
etime	所用时间的函数
etree	矩阵消元树结构
etreeplot	绘制消元路径
eval	运行包含 MATLAB 表达式的字符串
exist	检验变量或文件是否已经定义
exp	按元素的指数运算
expand	符号表达式展开

（续）

expint	指数积分
expm	矩阵指数函数
expm1	Expm()函数的.m 文件的实现
expm2	Taylor 级数求矩阵指数
expm3	特征值特征向量法求矩阵指数
eye	生成单位矩阵
ezplot	符号函数平面图形
factor	因式分解
fclose	关闭一个或多个文件
fegtl	从文件读入一行数据（忽略换行）
fegts	从文件读入一行数据（保留换行）
feof	测试文件是否结束
ferror	查询文件输入/输出的错误状态
feval	执行字符串指定的文件
fFeather	羽状图形绘制
fft	一维快速傅里叶变换
fft2	二维快速傅里叶变换
fftdem	快速 fourier 变换演示
fftshift	取消谱中心零位
fgetl	按行从文件中读取数据并清除文件定位器
fgets	按行从文件中读取数据并保留文件定位器
fieldnames	结构的属性名称
figure	建立图形窗口
fileparts	生成文件的路径、文件名和版本等部分信息
fill	绘制充填的二维多边形
fill3	绘制充填的三维多边形
filter	IIR 或 FIR 滤波
filter2	二维数字滤波
find	查找非 0 元素的下标和值
findstr	由一个字符串中查找
findsym	确定符号变量
finite	若参数为有限元素则为真
finverse	反函数
fitdemo	非线形最优化拟合演示
fix	对零方向取整数
flag	红白蓝黑基色颜色表
flipdim	数组翻转
fliplr	按左右方向翻转元素
flipud	按上下方向翻转矩阵元素
floor	对负无穷方向取整数
flops	计算浮点运算的次数
fmin	单变量函数求极小值
fmins	单变量函数求极小值
fminu	用 bfgs 法求无约束最优化问题
fopen	打开文件
for	指定循环次数的循环
format	设置输出格式

（续）

formula	返回在线函数对象的计算公式
fourier	Fourier 级数展开图形演示
fplor	一元函数极小点
fplotdemo	函数图形绘制演示
fprintf	有格式地向文件写入数据
fread	从文件读入二进制数据
frewind	将文件指针至文件开头
fscanf	从文件有格式地读入数据
fseek	设置文件定位器获取文件定位器的位置
ftell	获得文件位置指针
full	由稀疏矩阵变换常规矩阵
fullfile	生成文件的路径和文件名的全称
function	MATLAB 函数表达式的引导符
funm	计算一般矩阵函数
funtool	函数计算器
fwrite	将二进制数据写入文件
fzero	单变量函数求根
gallery	测试矩阵的类型
gamma	Gamma 函数
gammainc	Gamma 函数
gammaln	Gamma 函数
gca	获得当前坐标轴的句柄
gcd	最大公因子
gcf	获得当前图形的窗口句柄
gco	当前图形对象句柄
gestalt	苹果机上的启动函数
get	获得对象属性
getenv	获得环境参数
getfield	获取结构的属性
getframe	获得一幅“电影”图像
ginput	由鼠标器作图像输入
global	定义全局变量
gmres	标准化最小残差法
gplot	绘制图论图形
gradient	近似梯度计算
gray	线形灰度颜色表
graymon	将图形窗口设置成灰度默认值
grid	给图形加网格线
griddata	插值用数据网格生成
gtext	在鼠标指定的位置加文字说明
guide	图形用户界面设计向导
hadamard	生成 Hadamard 矩阵
hankel	生成 Hankel 矩阵
help	MATLAB 函数和 M 文件在线帮助
help uitools	图形用户界面工具函数
hess	求取 hessenberg 标准型
hex2dec	十六进制到十进制的转换

（续）

hex2num	把 16 进制转换为双精度
hidden	网格图隐含线实质开关
hilb	生成 Hilbert 矩阵
hilh	生成 hibert 矩阵
hist	直方图绘制
hold	当前图形保护模式
home	将光标移动到左上角位置
horner	嵌套多项式
hostid	MATLAB 服务器的主机代号
hot	黑红黄白基色颜色表
hsv	色度饱和值（HSV）颜色表
hsv2rgb	HSV 对 RGB 颜色的转换
humps	一元函数用例
i	虚数单位
if	条件转移语句
if	条件执行关键字
ifft	一维快速傅里叶变换
ifft2	二维快速傅里叶变换
ifourier	逆傅里叶变换
ilaplace	逆拉普拉斯变换
imag	求取虚部函数
image	创建图像
imagedemo	MATLAB 与 4.0 版天下处理功能演示
imfinfo	返回图形文件信息
imread	从图形文件中读出数据
imwrite	把图形数据定写入文件
ind2sub	数组的下标转换
inf	无穷大（保留变量）
inferioero	类关系运算符
info	显示 MATLAB 与 mathworks 信息
inline	创建 inline 对象
inmem	列出内容中的函数
inpolygon	检验一些点是否在多边形内
inpolygon	质数分解
input	带有提示的键盘输入函数
inputname	输入参量名称
int	符号积分
int2str	把整数转换为字符串
interp1	一维插值（一维查表）
interp2	二维插值（二维查表）
interp3	三维插值
interpft	利用 FFT 的一维插值
interpn	多维数据插值
interpn	多维插值
intersect	两个向量求交集
intro	MATLAB 引言信息
inv	矩阵求逆

（续）

inverse	符号逆
invhilb	生成逆 hibert 矩阵
ipermute	数组维数逆变换
isa	检验某变量是否为给定类的对象
iscell	检测是否 cell 数组
iscellstr	检测是否字符串 cell 数组
ischar	检测是否字符串数组
isempty	若参数为空矩阵，则结果为真
isequal	检测两数组是否相等
isfield	检测结构体数组域名
isfinite	检测是否有限值
isglobal	若参数为全局变量则为真
ishandle	检测是否图形句柄
ishold	若屏幕处于保护状态则为真
isieee	若有 IEEE 算术标准则为真
isinf	若参数为 inf，则结果为真
isletter	若字符创为字母组成则为真
islogical	检测是否逻辑数组
ismember	检测某元素是否属于某集合
isnan	若参数为 nan,则结果为真
isnumeric	检测是否数值数组
isppc	检测是否 macintosh power pc 机型
isprime	检测是否素数
isreal	检测是否实数
isspace	检测是否空字符
issparse	若矩阵为稀疏矩阵表示则为真
isstr	若参数为字符串，则结果为真
isstruct	检测是否结构体
isstudent	检测 matlab 是否学生版
isvms	检测 MATLAB 是否 VMS 版本
iztrans	逆 Z 变换
j	虚数单位
jacobian	符号雅可比矩阵
jet	HSV 色调的变化型
jordan	Jordan 标准型
keybord	启动键盘管理程序
knot	围绕三维结的柱形显示
kron	Kronecke 乘积函数
laplace	拉普拉斯变换
lasterr	查询最近一个错误信息
latex	将符号表达式转换为 LATEX 形式
lcm	最小公倍数
leastsq	用最小二乘法求解无约束最优化问题
legend	图例说明
legendre	Legendre 函数
length	查询向量的维数
life	Conway 生命假设的 MATLAB 版

（续）

light	光源
line	创建线对象
lines	由属于 ColorOrder 生成的色图矩阵
linsolve	线性代数方程组符号解
linspace	构造线形分布的向量
load	从文件中读入变量
log	自然对数函数
log10	常用对数函数
log2	以 2 为底的对数
logical	生成逻辑数组
loglog	全对数坐标图绘制
logm	矩阵的对数
logspace	生成等指数间隔的向量
lookfor	对 HELP 信息中关键词查找
lorenz	Lorenz 混沌吸引子的曲线
lower	将一个字符串内容转换成小写
lp	求解线性规划问题
lscov	已知协方差的最小二乘解
lu	矩阵的三角（LU 分解）
luinc	不完全 LU 分解
magic	生成魔方矩阵
map2sum	将 MAPLE 串转换为符号对象
maple	调用 MAPLE 系统
mat2str	将矩阵化为字符串
matlabrc	启动 MATLAB 的主程序
matlabroot	安装 MATLAB 的根目录
max	求向量中最大元素
mean	求向量各元素均值
median	求向量各元素中间值
membrane	产生 mathworks 公司标志
menu	为输入生成选择菜单
mesh	三维网格图形
meshc	带有等高线的网格图形
meshgrid	生成三维比数据点人 X 坐标矩阵和 Y 坐标矩阵
meshz	带有零平面的的三维网格图形
mextpow2	找出下一个 2 的指数
mfun	MAPLE 函数的数值计算
mfunlist	为 MFUN 指定函数
mhelp	MAPLE 核的帮助文件
min	求向量中最小元素
mod	求余
more	控制命令窗口的输出页项
movie	播放存储的“电影”幅面
moviein	初始化“电影”各幅图像内存
mpa	MAPLE 变量赋值
mprint	打印 MAPLE 变量
mu2lin	声音文件对线形标度文件的转换

（续）

nan(NaN)	不定式
nargchk	函数输入输出参数个数检验
nargin	函数中实际输入变量个数
nargin,nargout	函数的输入参数和输出参数个数
nargout	函数中实际输出变量个数
nchoosek	求组合数
ndgrid	为多维函数准备数据
ndgrid	为多维函数和插值勤生成数据
ndims	多维数组的维数
newplot	Newplot 特性的.m 文件前缀
nextpow	最相邻的 2 的幂
nnls	非负最小二乘解
nnz	非零元素个数
nonzeros	非零元素
norm	求矩阵的范数
normest	估算范数
notebook	启用 Notebook 系统
now	当前日期号
null	右零空间
nullspace	求矩阵 0 空间的基
num2str	将数值转换为字符串
numden	提取分子分母
numeric	将符号矩阵化为数值形式
numgrid	二维区域格点编号
nzmax	允许的非零元素存储空间
ode113	用可变阶方法解非 STIFF 方程
ode113	解微分方程
ode15s	用可变阶方法解 STIFF 方程
ode23	微分方程低阶数值解法
ode23p	微分方程低阶数值解法并画图
ode23s	用低阶方法截 STIFF 方程
ode45	微分方程高阶数值解法
odedemo	常微分方程演示
odefile	定义微分方程
odeget	获取选项结构数组的值
odeset	生成和修改选项结构数组的值
ones	产生元素全部为 1 的矩阵
options	最优化控制参数
orient	设置打印纸方向
orth	正交空间
otherwise	swith 结构的默认执行关键字
pack	整理工作空间内存
paren	各种括号的查询信息
pascal	生成 pascal 矩阵
patch	低级填充多边形绘制函数
path	设置或查询 MATLAB 的路径
pathtool	路径浏览器

（续）

pause	暂停函数
pcg	预条件共轭剃度法
pcode	产生 P-文件
pcolor	伪颜色绘图
peaks	两变量的峰值函数演示
penny	便士硬币的各个角度视图
perms	所有可能变换
permutr	数组重组
pi	圆周率
pie	二维饼图
pie3	三维饼图
pink	粉色色调颜色表
pinv	伪逆矩阵
planerot	求平面旋转矩阵
plot	线性坐标图形绘制
plot3	绘制三维线或点型图形
plotyy	对比制图
ploy2sym	将系数向量转换为符号多项式
pol2cart	把极坐标圆柱坐标转换为直角坐标
polar	极坐标图形绘制
poly	求已知根的多项式的表达式
polyalm	求矩阵多项式的值
polyarea	多边形的面积
polyder	多项式求导
polyeig	多项式的特征值问题
polyfit	多项式曲线拟合
polyval	多项式求值
polyvalm	多项式矩阵求值
pow2	以 2 为底的指数
primes	生成质数列表
print	打印图形或将图形存盘
printopt	建立打印机默认值
prism	光谱颜色表
procread	安装一个 MAPLE 过程
prod	对向量中各元素求积
profile	对 M-文件进行监督
punct	各种标点符号的查询信息
qmr	QUASI-Minimal 残差法
qp	求解二次规划问题
qr	矩阵的正交三角化（OR）分解
qrdelete	QR 分解中删除一列
qrinsert	在 QR 分解中加入列
qrth	求矩阵列子空间的正交基
qtwrite	把 QuickTime 动画文件写入磁盘
quad	低阶数值积分算法
quad8	高阶数值积分算法
quaddemo	自适应变步长数值演示

（续）

quake	Loma prieta 地震模型
quit	退出 MATLAB 环境
quiver	箭头图形
qz	广义特征值问题求解（QZ 算法）
rand	生成均匀分步随机数和随机矩阵
randn	生成高斯分步随机数和随机矩阵
randperm	随机置换向量
rank	求矩阵的秩
rat、rats	有理近似
rbbox	擦除框
rcond	LINPACK 倒数条件估计
real	求取实部函数
realmax	能表示的最大浮点数
realmin	能表示的最小浮点数
relop	各种关系符号的查询信息
rem	除法的余数
repmat	复制和编排矩阵
reset	恢复对象特征
reshape	中断函数运行
residue	求部分分式表达式
return	返回到主调函数的命令
rgb2hsv	RGB 对 HSV 颜色的转换
rgbplot	绘制颜色图
ribbon	去掉一个域
rmfield	删除结构的属性
rmpath	删除 MATLAB 的搜索路径
roots	求多项式的根
rose	极坐标（角度）直方图绘制
rosser	典型的对称矩阵特征值问题测试
rot90	将矩阵元素旋转 90 度
round	截取到最近的整数
rref	矩阵的行阶梯型实现
rrefmovie	消元法解方程过程演示
rsf2csf	把实的 Schur 形式转换为复的 Schur 形式
rsums	黎曼求和交互式演算
save	将工作空间中变量存盘
saxis	声音坐标轴处理
schur	Schur 分解
script	Matlab 语句及文件信息
script	命令文件
sec	正割函数
sech	双曲正割
semilogx	X 轴半对数坐标图形绘制
semilogy	Y 轴半对数坐标图形绘制
sepdemo	有限元网格图演示
set	设置对象属性
setdiff	求两个向量的差集

（续）

setfield	设置属性值
setstr	将数值转换为字符串
shading	阴影模式
shiftdim	维数移位
sigdemo1	离散 fourier 变换演示
sigdemo2	连续 fourier 变换演示
sign	符号函数
simple	最简形式
simplify	符号表达式化简
sin	正弦函数
singvals	符号异或值
sinh	双曲正弦
size	求数组的维数大小
slash	求解线性方程(左除右除)信息
slice	容量可视图形
solve	代数方程组符号解
sort	将元素按升序排列
sortrows	将行按升序排列
sound	将数据向量转换为声音
sounddemo	Matlab4.0 的声音功能演示
spalloc	为稀疏矩阵分配的存储空间
sparse	从常规矩阵转换稀疏矩阵
sparsity	稀疏矩阵排序效应演示
spaugment	建立最小二乘增广系统
spconvert	从外部格式中载入稀疏矩阵
spdemo	超二次锥面的显示
spdiags	生成稀疏带状矩阵
specular	反射
spenes	将原稀疏矩阵非零元素有 1 取代
speye	单位稀疏矩阵
spfun	非 0 元素的函数计算
spfunf	对稀疏矩阵处理的非线性函数
sph2cart	把球坐标转换为直角坐标
sphere	产生球面
spinmap	使颜色旋转
spline	立方插值
spline2rd	二维样条函数演示
spones	把矩阵中的非 0 元素全用 1 代替
spparms	设置稀疏矩阵程序的参数
sprand	均匀分布随机稀疏矩阵
sprandn	高斯分布随机稀疏矩阵
sprandsym	稀疏对称随机矩阵
sprank	结构秩数
spring	由紫红色和黄色生成的色图矩阵
sprintf	格式输出字符串
sprt	平方根函数
sprtm	矩阵的平方根

（续）

spy	稀疏矩阵的图形表示
sqrt	平方根
sqrtm	矩阵平方根
squeeze	删除所有的单维
sscanf	从字符串中读入格式数据
stairs	阶梯图形绘制
startup	启动 MATLAB 的自执行文件
std	求向量中各元素标准方差
stem	离散序列柄状图形绘制
stem3	三维火柴杆图
stercat	字符串组合
str2mat	字符串转换成矩阵
str2num	字符串转换为实型数据
strcat	字符串水平连接
strcmp	字符串比较
strings	关于 matlab 字符串的帮助信息
strjust	给出字符串最终结果
strmat	由字符串构造文本矩阵
strmatch	查找符合要求的行
strncmp	比较字符串的前 n 个字符
strrep	字符串查找和替换
strtok	查找某个字符最先出现位置
struct	生成结构数组
struct2cell	把结构数组转换为细胞数组
strvcat	字符串的竖向组合
sub2ind	单下标
subexpr	按子表达式改写
subplot	将图形窗口分成若干个区域
subs	变量代换
subscrbe	成为 matlab 的签约用户
subspace	子空间
sum	对向量中各元素求和
summer	由绿色和黄色生成的色图矩阵
sunspots	太阳黑子活动摸拟
superiorto	类关系运算符
surf	三维表面图形
surface	创建曲面
surfc	带有等高线的三维表面图形
surfl	带有光照阴影的三维表面图形
surfnorm	表面图形规范化
svd	奇异值分解(svd)
svds	少数奇异值
switch	switch 结构关键字
sym	构造符号对象建立和访问符号矩阵
sym2map	将符号对象转换为 MAPLE 串
sym2poly	将符号多项式转换为系数向量
symadd	符号加

（续）

symbfact	符号因式分解
symdiff	符号微分
symdiv	符号除
symmmd	对称最小度排序
symmul	符号乘
sympow	符号幂运算
symrcm	反向 Cughill-McKee 排序
syms	以简捷命令行形式构造符号对象
symsize	求符号矩阵行列数
symsub	符号减
symsum	符号求和
symvar	确定符号变量
symvars	符号变量代换
tan	正切函数
tanh	双曲正切
taylor	泰勒级数
tempdir	列出系统的临时目录
tempname	列出临时文件名
terminal	设置图形终端类型
text	在图形上加文字说明
tic	启动秒表计时器
title	给图形加标题
toc	秒表终止和显示
toeplitz	生成 Toepliz 矩阵
trace	求矩阵的迹
transpose	符号矩阵转置
trapz	梯形法求数值积分
treelayout	树状结构
treeplot	画出分割路径的图形
tril	提取矩阵的下三角部分
triu	提取矩阵的上三角部分
tsearch	搜索 Ddlaunay 三角形
type	显示文件内容
uicontrol	建立用户界面控制的函数
uimenu	创建用户界面菜单对象
uint8	转换为 8 位型
uisetcolor	标准颜色设置对话框
uisetfont	标准字体设置树对话框
ungetfile	标准读盘文件名处理对话框
union	求两个向量的并
unique	消除重复元素
unique	返回向量的元素值（无相同的值）
unix	执行操作系统命令并回结果
unwrap	除去每 360 度的跳跃
upper	将一个字符串内容转换为大写
vander	生成 vandermonde 矩阵
vap	变精度算法

（续）

varargin	可变长度的输入参数表
varargout	可变长度的输出参数表
vectorize	向量化表示
ver	显示程序版本号
version	显示 matlab 版本号
vibes	L 型振荡动画
view	三维图形视口指定
viewmtx	显示坐标变换矩阵
voronoi	Voronoi 图
vpa	求符号表达式值
warning	显示警告信息
waterfall	瀑布型图形
wavread	读 wav 声音文件
wavwrite	写 wav 文件
weekday	本周第几天
what	列出当前目录上的 M 文件 MAT 文件和 MEX 文件
whatsnew	显示 MATLAB 和工具箱的 README 文件
which	找出函数与文件所在的目录名
while	循环语句
whitebg	将图形窗口设置成白色背景
who	简要列出工作空间变量名
whos	列出工作空间中的变量详细内容
why	给出简要的回答
Wilkinson	wilkinson 特征值测试矩阵
winter	由蓝色和绿色生成的色图矩阵
wk1read	从 Lotus1-2-3WK1 电子表格文件中读出矩阵
wk1write	把矩阵写入 Lotus1-2-3WK1 电子表格文件
xlabel	给图形加 x 坐标说明
xlsetrange	把数据写入 Excel 表格
xor	逻辑异或
xpcalc	数字运算演示
xpgiv	GIVENS 变换演示
yesinput	带有询问提示和应答检查的输入
ylabel	给图形加 y 坐标说明
zerodemo	求根演示
Zeros	生成全 0 数组
zlabel	给图形加 y 坐标说明
zoom	图形缩放（放大缩小）
ztrans	Z 变换

附录 B　TOOLBOX 基本工具箱函数

MATLAB 系统提供的 TOOLBOX 工具箱，有内容极为丰富的函数用于自动控制方面。这里仅列出工具箱函数命令按英文字母的索引，以备用户查阅。关于函数的使用格式，请参见联机帮助。

函数名	函数功能
abcdchk	检测（A，B，C，D）组的一致性
abs	取绝对值（幅值）
accdemo	弹簧质量校准问题
acker	单输入单输出(SISO)系统极点配置
addmf	将求属度函数加到 FIS 中
addrule	将规则加到 FIS 中
addvar	将变量加到 FIS 中
anfis	Sugeno-typeFIS 的训练程序
angle	取幅角
append	将两个用状态空间表示的系统连接起来
ar	求 AR 模型的参数估计
are	求解 Riccati 代数方程
aresolv	广义连续时间 Riccati 方程求解
armax	求 ARMAX 模型的预测误差估计
arx	求 ARX 模型的最小二乘估计
arx2th	求 ARX 模型的 theta 格式
arxstruc	计算 ARX 类模型的损伤函数
attgoal	达到多目标
augss	系统增广（状态空间模型）
augstate	将系统状态增扩到输出方程中
augtf	系统增广（传递函数模型）
balmr	截断均衡模型简化
balreal	求状态空间的均衡实现
bandemo	香蕉型函数的极小化
barerr	每个输出适量的误差条形图表
bartlett	Bartlett(巴特利特)窗
besselap	Bessel 模拟低通滤波原型
besself	Bessel（贝塞尔）函数
bestblk	分块处理的最佳块大小
bilin	利用可异值分解变系统为状态空间系统
bj	求 Box – Jenkins 模型的预测误差估计
blackman	Blackman（布莱克曼）函数
blkbuild	从传递函数框图构成对角线状态方程结构

（续）

blkproc	按块处理一图像
blkrsch	通过 cschur 得到块有序实形式
bmpread	磁盘读 BMP（Microsoft windows 下的位图）文件
bmpwrite	从磁盘中读 GIF 文件
bode	Bode (波特)图（频域响应）
bodeplot	绘出传递函数或频谱的伯德图
boildemo	锅炉系统的 LQG 设计
boxcar	矩形窗
branch	从树中提取一分支
brighten	增强或削弱颜色板
bstschmr	相对误差 Schur 模型简化
buttap	Butterworth 模拟低通滤波原型
butter	Butterworth（比特沃思）滤波器设计
buttord	Butterworth 滤波器阶的选择
bwarea	二进制图像中的目标区域
bweuler	欧拉数
bwmorph	形态算子
bwperim	二进制图像中目标的周围
c2d	将连续系统离散化
c2dm	将带离散化方法选项的连续系统离散化
c2dt	将带输入纯滞后的连续系统离散化
canform	定义正则模型结构
canon	求系统状态空间的正则实现
canstart	求带初始参数估计的多变量 theta 模型
cceps	倒谱分析和最小相位重构
cgloci	连续特性增益轨迹
cheb2ap	Chebyshev II 型模拟低通滤波原型
cheb2ord	Chebyshev II 型滤波器阶的选择
cheblap	Chebyshev I Bessel 模拟低通滤波原型
cheblord	Chebyshev I 型滤波器阶的选择
chebwin	Chebyshev 窗
cheby1	chebyshev（切比雪夫）I 型滤波器设计
cheby2	chebyshev（切比雪夫）II 型滤波器设计
chop	取 n 个重要的位置
cloop	求单位反馈系统的闭环传递函数或状态方程
cmgamdef	缺省的 γ 校正表
cmgamma	γ 校正颜色板
cmpermute	转换颜色板位置
cmunique	寻找唯一的颜色板及相应的图像
cohere	相关函数平方幅值估计
col2im	重新排列以形成图像
colfilt	局部非线性滤波
colorbar	显示颜色条
colormap	设置或获取颜色查找表
combvec	创建所有的适量集
compare	将仿真的或预测的输出与测量输出相比较
compet	竞争层传递函数

（续）

coneddlg	管理 NCD 工具箱固定编辑器的对话框
connect	方框图建模
constr	约束极小化
contrnce	建立 NCD0 固定图形的用户界面控制
conv	求两个多项式的卷积
conv2	求二维卷积
convertm	变约束矩阵为最优化格式
convmtx	卷积矩阵
corr2	二维相关系数
corrcoef	相关系数矩阵
costfun	NCD 优化的代价函数
cov	协方差矩阵
covar	求连续系统对白噪声的协方差矩阵
covf	求数据矩阵的协方差函数估计
cplxpair	将复数归成复共轭对
cra	相关分析
cschur	通过复旋转得有序复 Schur 形式
csd	交叉谱密度（CSD）估计
ctrb	计算能控性矩阵
ctrbf	系统可控与不可控分解
ctrldemo	控制系统工具箱一般功能介绍
cubic	内插 4 点以找出极大值
cubici1	内插 2 点和梯度，以估计极小值
cubici2	内插 3 点和 1 梯度
cubici3	内插 2 点和梯度，以找出步长和极小值
cumsum3d	三维矩阵封装成二维矩阵时的累积和
curobj	提供有关当前点的信息
d2c	将离散系统连续化
d2cm	将带连续化方法选项的离散系统连续化
d2d	离散时间系统重采样
damp	求连续系统的自然频率和衰减因子
daresolv	广义离散时间 Riccati 方程求解
datdemo	数据拟合成曲线
dbalreal	求离散状态方程的均衡实现
dbode	绘制离散系统伯德图
dcgain	求连续系统的稳态增益
dcgloci	离散特性增益轨迹
dcovar	求离散系统对白噪声的协方差响应
dct	一维离散余弦变换
dctdemo	二维离散余弦变换图像压缩演示
dctmtx2	一元二维离散余弦变换矩阵
ddamp	求离散系统的自然频率和阻尼系数
ddcagin	求离散系统的稳态增益
decimate	降低序列的取样速率
deconv	反卷积和多项式除法
defuzz	去模糊隶属度函数
delaysig	从信号矩阵中建立退化的信号矩阵

（续）

delline	从 NCD 图中删除所有的图
deltalin	对 PURELIN 神经元的 δ 函数
deltalog	对 LOGSIG 神经元的 δ 函数
deltatan	对 TANSIG 神经元的 δ 函数
demod	通讯仿真中的解调
des2ss	线性分式变换
destim	从增益矩阵中形成离散状态估计器
det2	二维离散余弦变换
detrend	删除线性趋势
dexresp	离散取样响应函数
dfildemo	有恨精度滤波器设计
dfrqint	离散 Bode 图的自动定范围的算法
dfrqint2	离散 Nyquist 图的自动定范围的算法
dgram	求离散系统的能控性和能观性检测矩阵
dh2lqg	离散时间 H2 综合
dhinf	离散时间 H∞综合
dialog	主对话框建立M文件
dilate	加浓二进制图像
dimpulse	求离散系统单位脉冲响应
dinitial	求带初始条件的离散系统时域响应
dintdemo	双积分器系统的 H∞设计
diric	产生 Dirichlet 或周期 sinc 函数
diskdemo	硬盘控制器的数字控制设计
dist	计算矢量距离
distsl	到直线间的距离
dither	Floyd – Steinberg 图像颤抖算法
ditherc	图像颤抖的 MEX 文件
dividecb	将固定界分为两部分
dlqe	设计离散 Kalman 滤波器
dlqew	通用离散 Kalman 滤波器设计
dlqr	设计离散线性二次型调节器
dlqry	设计离散线性二次型输出调节器
dlsim	对任意输入的离散系统进行仿真
dlyap	求解离散系统的 Liapunov 方程
dmodred	将离散模型降阶
dmulresp	离散多变量响应函数
dnichols	绘制离散系统 nichols(尼科尔斯)图
dnyquist	绘制离散系统 Nyquist(奈魁斯特)图
dode	绘制 dode(伯德)图
donep	收回 Close 按钮和菜单
dreg	构成离散 LQG 控制器
dric	离散 Riccati 方程留数计算
driccond	离散时间 Riccati 方程的条件数
drmodel	产生 n 阶随机离散模型
drss	产生 n 阶随机离散系统
dsigma	绘制离散奇异值 Bode 图
dsigma2	DSIGMA 实用工具函数

（续）

dsigmf	两个"s"型隶属度函数的差
dsort	按幅值对离散系统的复数极点排序
dss	生成状态空间模型
dssdata	获得状态空间模型数据
dstep	求离散系统阶跃响应
dtimvec	离散时间响应的自动定范围算法
dtrend	从数据集中删除方位
edge	边界提取
eig	计算系统极点
eigfun	返回分类特征值的函数
elem3d	三维散阵封装成二维矩阵的元素位置
elimone	消去一变量
ellip	椭圆滤波器设计
ellipap	椭圆模拟低通滤波器原型
ellipord	椭圆滤波器阶的选择
erode	冲淡二进制图像
erroncd	管理 NCD 产生的常见错误，它调用 errordig(出错对话框)
errordlg	建立出错对话框
errsurf	计算误差曲面
esort	按实部对连续系统极点排序
estim	构成系统状态估计器或观测器
etfe	求实验传递函数和周期图
evalfis	完成模糊推理计算
evalfr	计算系统单频率点的频率响应
evalmf	隶属度计算
exresp	取样响应函数
fbode	绘制快速伯德图
fcm	利用模糊 C 平均聚集方法找出簇
feedback	将两个系统反馈连接
ffplot	绘出传递函数或频谱的频率特性图
fft2	二维快速付里叶变换
fftshift	重新排列 FFT 的输出
figflag	当图形为当前显示在屏幕上时，其值为真
fillaxes	建立约束边界并进行数据检测
filt	离散系统求 z^{-1}(D S P 形式)的传递函数
filter	对 SISO 系统的 z 变换进行仿真
filter2	二维数字滤波器
filtfilt	零相位数字滤波
filtfun	频率响应和根
filtfun2	频率响应范数和根
filtic	filter 函数初始条件选择
findmax	在数据向量内插极大值
findmax2	在数据矩阵中内插极大值
fir2	基于窗函数的 FIR 滤波器设计——任意响应
firdemo	二维 FIR 滤波器演示
firl	基于窗函数的 FIR 滤波器设计——标准响应
firls	最小二乘 FIR 滤波器设计

（续）

fitfun	返回拟合数据中的误差范数
fitfun2	返回拟合数据中的误差矢量
fixpa	更新状态空间或 ARX 模型结构的确定参数值
fixpar	在状态空间和 ARX 模型结构中找出要修正的参数
fltfilt	重叠相加法 FFT 滤波器实现
fmin	无约束极小化(标量情况)
fmins	利用单纯形搜索的无约束极小化
fminu	利用梯度搜索的无约束极小化
foptions	参数设置
forceit	在已存在的界限内插入一子集
freqint	Bode 图的自动定范围算法
freqint2	Nyquist 图的自动定范围算法
freqresp	计算系统频率响应
freqs	求拉氏变换频率响应
freqspace	二维频率响应的频率空间
freqz	求 Z 变换频率响应函数
freqz2	二维频率响应
fsamp2	通过频率取样的二进制 FIR 滤波器设计
fsolve	非线性方程求解
fspecial	特殊的二维滤波器
ftrans2	通过频率变换的二维 FIR 滤波器设计
fuzzy	基本 FIS(模糊推理系统)编辑器
fwind1	使用一维窗函数的 FIR 滤波器设计
fwind2	使用一维窗函数的 FIR 滤波器设计
gauss2mf	双边高斯曲线隶属度函数
gaussmf	高斯曲线隶属度函数
gbellmf	广义钟形隶属度函数
genfis1	利用一般方法产生 FIS 矩阵
genfis2	利用减法聚集法产生 FIS 矩阵
gensig	信号生成器
gensurf	产生 FIS 输出曲面
get	获得 LTI 对象的属性值
getff	选择用于绘图的频率函数
getfis	获得模糊系统的特性
getimage	从坐标系中读取图像数据
getline	利用橡皮线跟踪鼠标移动
getmfth	获取定义模型结构的 M 文件的文件名
getncap	可得到获取模型的数据点数和参数个数
getpts	利用可视点跟踪鼠标移动
getrect	利用橡皮矩形跟踪鼠标移动
gett	为模型获取取样间隔
getzp	在由 th2zp 函数产生的零极点格式中，提取零点和极点
gif	压缩 GIF 数据
gifwrite	将 GIF 文件写入到磁盘
givens	旋转
goalfun	目标达到问题转换成约束条件问题
goalgra	变换目标达到问题中的梯度

（续）

graderr	用于检查梯度的不一致性
graft	从树中增加一分支
gram	求连续系统的能控性和能观性 gramian 矩阵
gray	线性灰度颜色板
gray2ind	变灰度图像为附标图像
grayslice	密度（强度）限幅
grpdelay	平均滤波延迟（群延迟）
h2lqg	连续时间 H2 综合
hamming	Hamming（哈明）窗
hardlim	硬限幅传递函数
hardlims	对称硬限幅传递函数
hav2rgb	变 HSV 值为 RGB 颜色空间
hdfpeek	在 HDF 文件中列出目标标记 / 参考对
hdfpeekc	搜索 HDF 文件的 MEX 文件
hdfread	从 HDF 文件中读取数据
hdfreadc	读 HDF 文件的 MEX 文件
hdfwc	写 HDF 文件的 MEX 文件
hdwrite	写出数据到 HDF 文件中
helpdlg	显示一帮助对话框
hilbert	Hilbert (希尔伯特)变换
hinf	连续时间 H∞综合
hinfdemo	飞机或大型窨结构的或 H∞设计示例
hinforpt	H∞综合的 γ 迭代
hintonw	绘制权值图
hintonwb	绘制权值和偏差图
histeq	直方图均衡化
hotkey	热键帮助
housh	构造 Householder 变换
hsv,hot,jet	颜色板
hsv2rgb	变 HSV 值为 RGB 值
idct	一维逆离散余弦变换
idct2	二维逆离散余弦变换
idfilt	通过 Butterworth 滤波器对数据进行滤波
idplot	绘出输入— 输出关系图
idsim	仿真给定系统
idsimsd	通过模型响应仿真说明不确定性
ifft	一维逆快速付里叶变换矩阵
ifft2	二维逆快速付里叶变换
im2bw	变图像为黑白图像
im2col	重新排列成列
im2gray	变图像为灰度
imadjust	调整并增强图像强度
image	显示附标图像
imagesc	数据定标并按图像显示
imapprox	利用较少颜色的图像逼近加标图像
imargin	利用内插技术求增益和相位裕度
imcontour	图像等高线

（续）

imcrop	修剪图像
imdemo	一般图像处理演示
imhist	图像直方图
imhistc	图像直方图计算的文件
immovie	制作图像动画
imnoise	图像噪声
imp2ss	从脉冲响应到状态空间实现
impixel	一像素点的颜色
improfile	轮廓强度
impulse	求脉冲响应
impz	数字滤波器的冲激响应
imresize	改变图像大小
imrotate	旋转图像
imshow	显示所有类型的图像数据
imslice	在图像中获取 / 置入图像块
imzoom	放大或缩小图像和二维图形
ind2gray	变附标图像为灰度图像
ind2rgb	变附标图像 RGB 为图像
ind2vec	变下标适量为稀疏矩阵表示
inite	竞争层初始化
initelm	Elman 递归网络初始化
initff	至多三层的前向网络初始化
initial	计算连续时间零输入响应
initlin	线性层初始化
initlvq	LVQ 网络初始化
initp	感知层初始化
initsm	自组织映射初始化
interc	一般多变量内连系统
interp	提高取样速率（内插）
interp2	二维数据内插
intfilt	内插 FIR 滤波器设计
invfreqs	模拟滤波器拟合频率响应
invfreqz	离散滤波器拟合频率响应
iofr	内外因子分解（行类型）
isbw	当图像为黑白时，其值为真
isct	判断 LTI 对象是否为连续系统，若是返回 1 否则返回 0
isdt	判断 LTI 对象是否为离散系统，若是返回 1 否则返回 0
isempt	判断 LTI 对象是否为空，若是返回 1 否则返回 0
isgray	当图像为灰度时，其值为真
isind	当图像为加标图像时，其值为真
isproper	判断 LTI 对象是否为特定类型对象，若是返回 1 否则返回 0
issiso	判断 LTI 对象是否为 SISO 系统，若是返回 1 否则返回 0
issystem	辨识一系统变量
istree	辨识一树型变量
iv4	求 ARX 模型的近似最优辅助变量估计
ivar	求标量时间序列的 AR 部分的辅助变量估计
ivstruc	单输出类的输出误差拟合

（续）

ivx	求 ARX 模型的辅助变量估计
jetdemo	喷气式飞机偏航阻尼的典型设计
kaiser	Kaiser 窗
kalman	系统的 Kalman 滤波器设计
kalmd	连续系统的离散 Kalman 滤波器设计
kalmdemo	Kalman 滤波器设计与仿真
keyncd	NCD 按键函数
lab2sert	变标号为字符串
layout	定义对话框布局参数的底稿文件
learnbp	反向演播学习规则
learnbpm	带预测的反向演播学习规则
learnh	学习规则
learnhd	退化的学习规则
learnis	内星学习规则
learnk	学习规则
learnlm	学习规则
learnlvq	学习矢量量化规则
learnos	外星学习规则
learnp	感知层学习规则
learnpn	归一化的感知层学习规则
learnwh	学习规则
leastsq	非线性最小二乘
levinson	Levinson – Durbin 递归算法
lftf	扇形变换
loadncd	装入并显示 NCD 数据
lofc	内外因子分解（列类型）
loglig	对称 S 型传递函数
lp	线性规划
lp21p	低通到低通模拟滤波器变换
lp2bp	低通到带通模拟滤波器变换
lp2bs	低通到带阻模拟滤波器变换
lp2hp	低通到高通模拟滤波器变换
lpc	线性预测系数
lqe	求 Kalman 滤波器的参数（线性二次型估计器设计）
lqe2	用 Schur 法求 Kalman 滤波器（线性二次型估计器）的参数
lqed	用连续系统的性能指标函数设计离散的 Kalman 滤波器
lqew	设计通用 Kalman 滤波器（线性二次型估计器设计）
lqg	LQG 最优控制综合
lqgreg	根据 kalman 估计器增益与状态反馈增益建立 LQG 调节器
lqr	设计线性二次型调节器
lqr2	用 Schur 方法设计线性二次型调节器
lqrd	用连续系统的性能指标函数来进行离散二次型调节器设计
lqry	设计线性二次型输出调节器
lsim	对任意输入的连续系统进行仿真
lsint	初始化无约束极小二乘程序的函数
ltifr	求线性定常系统的频率响应
ltiview	LTI 线性时不变系统观测器

（续）

ltrdemo	LQR/LTR 设计示例：飞机
ltru	LQG 闭环传递补偿
ltry	LQG 闭环传递补偿
lv	求单输出 ARX 模型的辅助变量估计
lyap	求解连续系统 Liapunov 方程
lyap2	用特征值分解方法求解 Liapunov 方程
mainncd	一般 NCD 帮助
makesurf	建立并限界曲面
margin	求幅值和相角裕量及幅值和相位交界穿越频率
mat2gray	变矩阵为（灰度）图像
matlabrc	MATLAB 的主启动 M 文件
mean2	矩阵的均值
medfilt1	一维中值滤波
medfilt2	二维持中值勤滤波
menuncd	建立 NCD 固定图形的用户界面菜单
mf2mf	在函数之间变换参数
mf2th	将用户定义的模型结构转换成 theta 格式
mfedit	隶属度函数编辑器
mfilter2	屏蔽滤波
midpoint	产生中点值
minimax	极小极大求解
minipars	NCD 最小化分析
minreal	求状态方程的最小实现
mksys	为系统建立一树型变量
modred	将模型降阶
modstruc	构造用于 ms2th 命令的模型结构
modulate	通讯仿真中的调制
montage	按矩形剪辑方式显示图像
montevar	初始化 Monte Carlo 仿真
mrdemo	鲁棒模型简化示例
mri.mat	人体心脑的磁性共振图像
ms2th	将标准的状态空间参数形式转换成 theta 格式
mudemo	μ 综合示例
mudemo1	μ 综合示例
mulresp	多变量响应函数
muopt	具有实／复数混合不确定性系统的 SSV(结构化奇异值)上界
myqplot	传递函数的 Nyquist 图
nargchk	检测M文件的变量数
nbdist	使用适量距离的邻域阵
nbgrid	使用栅格距离的邻域阵
nbman	使用 Manhattan 距离的邻域阵
ncd2init	为 ncddemo2 的优化进行设置
ncd3init	为 ncddemo3 的优化进行设置
ncd4init	为 ncddemo4 的优化进行设置
ncdblock	包含 NCD 框图的 SIMULINK 系统
ncddemo	包含所有 NCD 演示示例的 SIMULINK 系统
ncddemo1	PID 控制器

（续）

ncddemo2	带前馈控制器的 LQR
ncddemo3	多输入多输出的 PI 控制器
ncddemo4	倒摆演示
ncdglob	定义 NCD 全局变量
ncdlinit	为 ncddemo1 的优化进行设置
ncdtut1	控制设计示例
ncdtut2	系统辨识示例
ndx3d	三维数阵封装成二维矩阵的索引
newfis	产生新的 FIS
ngrid	为 Nichols（尼科尔斯）图绘制栅格线
nichols	绘制 Nichols（尼科尔斯）图
nlfdemo	二维非线性滤波演示
nlfilter	局部非线性滤波
nlinopt	执行优化算法
nnls	非负最小二乘
normc	归一化矩阵列
normh2	计算 H2 范数
normhinf	计算 H∞范数
normr	归一化矩阵行
ntsc2rgb	变 NTSC 值为 RGB 值
nwlog	对 LOGSIG 神经元产生 Nguyen-Widrow 随机数
nwtan	对 TANSIG 神经元产生 Nguyen-widrow 随机数
nyquist	绘制 Nyquist（奈魁斯特）图
obalreal	有序均衡实现
obsv	计算能观性矩阵
obsvf	系统可观与不可观分解
oe	求输出误差模型的预测误差估计
ohklmr	最优 Hankel 极小化逼近
optblock	打开一个 NCD 图形的底稿文件
optdemo	演示菜单
optfig	建立一个 NCD 固定图形
optint	初始化无约束极小化程序的函数
ord2	产生二阶系统的 A、B、C、D
osborne	通过 Osborne 法求得的 SSV 上界
pade	求延滞τ的 n 阶 pade 逼近式
parallel	将两个系统并联连接
paramdlg	管理 NCD 最优化参数对话框图的帮助
parsrule	分析模糊规则
pe	计算预测误差
pem	求通用参数模型的预测误差估计
penddata	为 ncdtut2（即倒摆）进行设置
perpxy	寻找最近的正交点
perron	计算 Perron 特征值
pcxread	从磁盘中读 PCX 文件中
pcxwrite	将 PCX 文件写入磁盘
pimf	π 形隶属度函数
place	求 MIMO 极点配置

（续）

platpv	绘出限幅神经元的感知器分类
plocfind	找出给定的一组根对应的根轨迹增益
plocus	绘制根轨迹图
plotep	在误差曲面上绘制权和基位置图
ploterr	绘出网络误差与时间的关系
plotes	绘制误差曲面图
plotfa	绘出目标模式及网络函数的逼近
plotfis	显示 FIS 输入/输出图
plotmf	显示出一个变量的所有隶属度函数
plotsm	绘制自组织映射图
plottr	绘出网络误差记录及自适应学习速率
plotvec	用不同颜色绘制矢量
pnormc	伪归一化矩阵列
pole	计算系统极点
poly	变根植表示为多项式表示
poly2rc	从多项式系数中计算反射系数
poly2str	变多项式为字符串
poly2th	从给定多项式构造 theta 格式矩阵
polystab	稳定多项式
predict	M 步超前预测
presen	在屏幕上显示参数模型
present	屏幕上的参数模型
printmat	带行列号打印矩阵
printopt	设置打印选项
printsys	打印出系统状态方程或传递函数
prony	利用 Prony 方法 的离散滤波器拟合时间响应
psd	信号功率谱密度（PSD）估计
psigmf	两个“s”形隶属度函数的积
psv	Perron 特征结构的 SSV
purelin	线性传递函数
pzmap	绘制线性连续系统的零极点图
qp	二次规划
qre	代数方程求解
quad2	内插 3 点以找出极大值
quadinter	内插 4 点以估计极小值
quant	离散化成某数值的整数倍
questdlg	建立提问对话框
radbas	径向基传递函数
radon	Radon 变换
randnc	产生归一化行随机数
randnr	产生归一化列随机数
rands	产生对称随机数
rangedlg	管理坐标系范围的对话框
rarmax	对 ARMAX 模型递归估计值
rarx	对 AR 模型递归计算估值
rbj	对 Box - Jenkins 模型递归计算估值
rc2poly	从反射系数中计算多项式系数

（续）

rceps	实倒谱和最小相位重构
rctdemo	鲁棒控制工具箱演示——主菜单
readfis	从磁盘中装入 FIS
readncd	与 README.M 文件内容相同
refdlg	管理 NCD 参考信号的对话框
refresho	使约束矩阵与图形一致
reg	构成连续 LQG 调节器
releview	规则观察器及模糊推理框图
resample	提高取样速率（内插）
resid	计算和预测与模型相关的残差
residue	部分分式展开
residuez	Z 变换部分分工展开或留数计算
rgb2gray	变 RGB 图像或值为灰度图像或值
rgb2hsv	变 RGB 值为 HSV 值
rgb2im	变 RGB 图像为附标或强度图像
rgb2ind	变 RGB 图像为附标图像
rgb2ntsc	变 RGB 值为 NTSC 值
rgbplot	绘制 RGB 颜色板分量的图形
ric	Riccati 方程留数计算
riccond	连续时间 Riccati 方程的条件数
rle	压缩编码数据
rlocfind	交互式地确定根轨迹增益
rlocus	画根轨迹
rltool	系统根轨迹设计器
rmmf	从 FIS 删除隶属度函数
rmodel	产生稳定的 n 阶随机连续时间模型
rmvar	从 FIS 中删除变量
roe	对输出误差模型递归计算估值
roicolor	用颜色定义感兴趣的区域
roipoly	定义感兴趣的多边区域
rpem	求通用参数模型的递推估计
rplr	对一般模型递归计算估值
rschur	Schur 模型简化
rss	产生稳定的 n 阶随机连续时间系统
ruleedit	规则编辑器及（句法）分析程序
satlins	对称饱和线性传递函数
saveload	当文件是从 SelectFile 中选择时，其值为真
sawtooth	产生锯齿波或三角波
schord	有序 Schwr 分解
searchq	线性搜索程序
sectf	稳定和逆稳定映射
segment	进行数据分段并跟踪快变系统
selstruc	根据各种准则选择模型结构
semifun	半定问题转换成约束问题
seminf	半定极小化
series	将两个系统串联连接
set	设置或修改 LTI 对象的属性值

（续）

setfis	设置模糊系统特性
sett	设置 theta 格式中的采样间隔
sfl	左边频谱分解
sfr	右边频谱分解
sgrid	在 s 平面上绘制连续系统根轨迹的 ω_n，Z 的格线
showfis	显示带注释的 FIS
showrule	显示 FIS 规则
sigma	绘制线性连续系统的奇异值 Bode 图
sigma2	SIGMA 工具函数
sigmf	“sigmoid(S)” 形隶属度函数
simuc	竞争层仿真
simuelm	Elman 递归网络仿真
simuff	前向网络仿真
simuhop	Hopfield 网络仿真
simulin	线性层仿真
simup	感受知层仿真
simurb	径向基网络仿真
simusm	自组织映射仿真
sinc	产生 sinc 或 $\sin(\pi t)/\pi t$ 函数
size	计算系统维数
slowfast	变传递函数模型为状态空间模型
smf	“s” 形隶属度函数
snapncd	以 22.5℃间隔排出约束条
solvehop	设计 Hopfield 网络
solvelin	设计线性网络
solverb	设计径向基网络
solverbe	设计精确的径向基网络
sos2ss	变系统二阶分割形式为状态空间形式
sos2tf	变系统二阶分割形式为传递函数形式
sos2zp	变系统二阶分割形式为零极点增益形式
spa	频谱分析
specgram	频谱分析
square	产生方波
ss	生成或转换成状态空间模型
ss2sos	变系统状态空间形式为二阶分割形式
ss2ss	求相似变换
ss2tf	将状态空间模型转换成传递函数模型
ss2zp	将状态空间模型转换成零极点增益模型
ssbal	状态空间的对角均衡实现
ssdata	获得状态空间模型数据
ssdelete	从给定状态方程中删除指定的输入序列和输出序列
ssselect	从大系统中选择子系统
ssv	结构化奇异值图
stabproj	慢 / 快分解
std2	二维标准差
step	求阶跃响应
stepdlg	阶跃响应对话框的帮助

（续）

stepfun	求阶跃函数
stmcb	利用 Steiglitz – MeBride 迭代方法求线性模型
str2mat2	变一行字符串为多行字符串
struc	求用于 arxstruc 和 ivstruc 命令的典型结构矩阵
subclust	利用减法聚集方法估算簇中心
subimage	显示多个图像
sumsqr	平方和
surfview	输出曲面观测器
tansig	正切 S 型传递函数
texted	收回 Port 可编辑的文本
tf	生成或转换传递函数模型
tf2ss	将传递函数模型转换成状态空间模型
tf2zp	将传递函数模型转换成零极点增益模型
tfchk	检测传递函数的一致性
tfdata	获得传递函数模型数据
tfe	从输入输出中估计传递函数
th2arx	将 theta 格式模型转换成 ARX 模型
th2ff	计算模型的频率函数以及标准偏差
th2par	将 theta 格式转换成参数和协方差矩阵
th2poly	将 theta 格式转换成相关多项式
th2ss	将 theta 格式转换成状态空间模型
th2tf	变 theta 格式为传递函数表示
th2zp	计算模型的零极点、稳态增益及标准差
thc2thd	将连续模型转换成离散模型
thd2thc	将离散模型转换成连续模型
the2thd	变连续时间模型为离散时间模型
thinit	保证预测器或系统稳定的基础上给 theta 参数赋随机初始值
tiff	压缩 tiff 编码数据
tiffread	从磁盘中读 TIFF 文件
tiffwrite	将 TIFF 文件中写入磁盘
timvec	连续时间响应的自动定范围算法
toptim	最优化测试组
toptimf	最优化测试组的测试函数
toptimg	最优化测试组的测试函数梯度
trainbp	利用反向演播训练前向网络
trainc	训练竞争层网络
trainelm	训练 elman 递归网络
trainlvq	训练 LVQ 网络
trainp	利用感知规则训练感知层
trainpn	利用归一化感知规则训练感知层
trainsm	利用 kohlnen 规则训练自组织映射
trainwh	利用 widrow hoff 规则训练线形层
trairbpx	利用快速反向演播训练网络
trapmf	梯形隶属度函数
tree	建立树变量
trees.mat	树的扫描图像
triang	三角窗

（续）

trimf	三角形隶属度函数
truesize	改变图像大小使之具有实际尺寸
tutdemo	启动教程
tzero	求传输零点
tzero2	求系统传输零点
tzreduce	在计算过零点时简化系统
uiguide	有关用户界面约定 / 标准 / 建议的说明
uncerdlg	不确定性变量对话框的帮助
undoncd	放弃上次 NCD 图形用户界面的操作
unfixpar	更新状态空间模型或 ARX 模型结构中的待估计参数
updatdlg	更新 NCD 对话框
v2sort	分类两向量，然后删去丢失的元素
vco	电压控制振荡器
vect2ind	变稀疏矩阵表示为下标矢量
vmquant	与彩色量化 MEX 文件接口的 M 文件
vrsys	返回标准系统变量名
vsort	匹配两根轨迹的向量
waitbar	显示等待条
warndlg	建立警告对话框
warp	将图像卷成曲面
wiener2	自适应二维维纳滤波
writefis	在磁盘中保存 FIS
xcorr	交叉相关函数估计
xcorr2	二维交叉相关
xcov	交叉协方差函数估计
xwdread	从磁盘中读 XWD 文件
xwdwrite	将 XWD 文件中写入磁盘
youla	Youla 参数化
yulewalk	递归数字滤波器设计
zgrid	在 z 平面上绘制离散系统根轨迹的 ω_n，Z 的网格线
zmf	“Z”形隶属度函数
zp2sos	变系统零极点增益形式为二阶分割形式
zp2ss	将零极点增益模型转换成状态空间模型
zp2tf	将零极点增益模型转换成传递函数模型
zpk	建立或转换零极点增益模型
zpkdata	获得零极点增益模型数据
zplane	离散系统零极点图
zpplot	零点和极点